The Analog Mixed Signal Company

ZWIR4501

Data Sheet and User Manual

Version: 1.3

Release Date: 19. August 2009

Copyright © 2009, ZMD AG,

All rights reserved. The material contained herein may not be reproduced, adapted, merged, translated, stored, or used without the prior written consent of the copyright owner. The information furnished in this publication is preliminary and subject to changes without notice.

Notes

Copyright © 2009, ZMD AG,

All rights reserved. The material contained herein may not be reproduced, adapted, merged, translated, stored, or used without the prior written consent of the copyright owner. The information furnished in this publication is preliminary and subject to changes without notice.

ZWIR4501 Data Sheet and User Manual (v 1.1)

Single-Chip 900 MHz RF Transceiver with Integrated Thin HW-MAC

IEEE 802.15.4 Compliant / ZigBee™ Ready

Description

The ZWIR4501 is a fully integrated system-on-chip CMOS transceiver providing license-free multi-channel operation in the 868.3 MHz (EU) and 902 MHz to 928 MHz (North America) ISM bands. This low-power RF transceiver is optimized for data rates up to 40 kbit/s and incorporates direct sequence spread spectrum technology (DSSS) to ensure reliable data transfer in hostile RF environments. The high level of integration, shown below, includes a thin Media Access Control (MAC) layer, resulting in a minimum of external components and lower application costs.

Applications

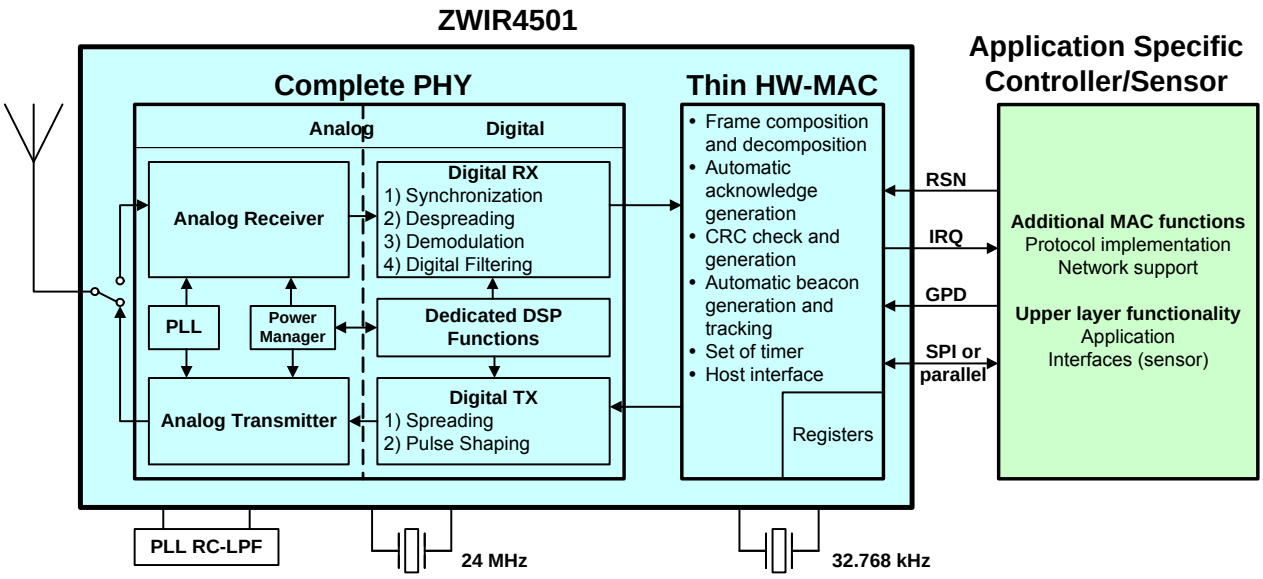
- Home Control
- Building Automation
- Remote Metering
- Remote Control
- Wireless Sensor Networks
- Industrial Networks
- Remote Keyless Entry (two-way)
- Health Monitor Networking
- PC/PDA Peripherals
- Consumer Electronics

Key Features

- IEEE 802.15.4 compliant / ZigBee™ ready
- ISM band transceiver with RF and baseband
- Integrated compliant PHY and Thin HW-MAC
- Direct Sequence Spread Spectrum (DSSS)
- Burst data rate 20 kbit/s (EU), 40 kbit/s (US)
- Transmit range 100+ meter (0 dBm, LoS)
- Low-power modes for battery-powered devices
- SPI and parallel interfaces
- 48-pin QFN (7 x 7 mm) package

Operating Data

Temperature Range	-40°C to +85°C
Supply Voltage, Core	+2.4 V
Supply Voltage, Digital I/O	+3.3 V
Supply Current, Tx active (0 dBm)	28 mA
Supply Current, Rx active	29 mA
Supply Current, Off Mode	1.3 µA
Frequency Bands	868 MHz and 915 MHz



Copyright © 2009, ZMD AG,

All rights reserved. The material contained herein may not be reproduced, adapted, merged, translated, stored, or used without the prior written consent of the copyright owner. The information furnished in this publication is preliminary and subject to changes without notice.

Table of Contents

1 General Device Description 6
1.1 Introduction 6
1.2 Features 6
1.3 RF Frontend Description 7
1.4 Pin Assignment 8
1.5 Pin Description 8
1.6 System Performance Summary 10
2 Device Specification 11
2.1 Recommended Operating Conditions 11
2.2 Absolute Maximum Ratings 11
2.3 Digital I/O 11
2.4 Operation Modes and Current Consumption 12
2.5 Startup Time 12
2.6 RF Parameter Summary 13
3 Application Circuit and External Components 14
3.1 Crystal Oscillator 15
3.1.1 Reference Crystal Oscillator (24 MHz) 15
3.1.2 Low Power Crystal Oscillator (32.768 kHz) 15
3.2 RF Phase Locked Loop 16
3.3 Antenna Connection and Power Amplifier Configuration 17
3.4 Clock Output (CLKO) 17
3.5 Interfacing 18
3.5.1 Power Up 19
3.5.2 Reset (RSN) 19
3.6 Synchronous Serial Peripheral Interface (SPI) 20
3.7 Parallel Interface 22
4 Integrated HW-MAC 23
4.1 Overview 23
4.2 Block Diagram 24
4.3 MAC Control 24
4.4 Interrupts 25
4.5 Frame Handling 26
4.6 Link Quality Indicator (LQI) 28
4.7 Receive Signal Strength Indicator (RSSI) 29
4.8 Energy Detection Level 30
4.9 Frame Filtering 31
4.10 Operation Mode Overview 32
4.11 Operation Mode Description 36
4.11.1 Low Power Modes 37
4.11.2 Transmit Mode (Tx) 39
4.11.3 Receive Mode (Rx) 44
4.11.4 Scan Modes 47

4.11.5 Automatic Beacon Generation 49
4.11.6 Beacon Tracking 52
4.11.7 Superframe Configuration 55
4.12 Timing Correction 56
4.12.1 Network Timing Correction 56
4.12.2 Beacon Interval Timing Correction 56
4.12.3 Superframe Timing Correction 58
4.13 General Purpose Timer Function 59
4.13.1 Overview 59
4.13.2 General Purpose Interrupt 60
4.13.3 Timer Controlled MAC Control Command Execution 60
4.13.4 Receive and Transmit Timestamp 60
4.14 Message Sequence Charts (MSCs) 61

5 Registers 71
5.1 Register Summary 71
5.2 IRQ Control, Configuration and Status 75
5.3 PHY Register 78
5.4 MAC Operating Control 81
5.5 MAC FIFO Register 83
5.6 MAC Tx Control 85
5.7 MAC Rx Control 91
5.8 MAC Ack Control 94
5.9 MAC Scan Control 97
5.10 MAC Beacon Control 99
5.10.1 MAC Beacon Generation Control 99
5.10.2 MAC Beacon Tracking Control 101
5.11 MAC Timer Control and Values 106
5.12 MAC Frame Filter Control 109
5.13 MAC Superframe and GTS Control 111
5.14 MAC CSMA Control 114
5.15 SPI Registers 116
5.16 CLKO Configuration 118
5.17 Recommended Startup Register Setup 119

6 Transmitter RF Spectrum Test Modes 120

7 Mechanical Specifications 121

8 Limitations 122

9 Document Revision History 122

10 List of Abbreviations 123

11 References 123

Important Note:

Radio transceivers and transmitters are regulated by international and national regulations. It is the users responsibility to be aware of and comply with the regulations of the telecommunications authority for the country where this equipment is to be used.

1 General Device Description

1.1 Introduction

The ZWIR4501 is a fully integrated CMOS transceiver, providing license-free multi-channel operation in the 868.3 MHz (for Europe) and 906 MHz to 924 MHz (for North America, US) ISM bands. This low power baseband transceiver is optimized for data rates up to 40 kbit/s (US) and incorporates Direct Sequence Spread Spectrum (DSSS) technology to assure reliable data transfer in hostile RF environments. The transceiver is highly integrated and includes a thin Medium Access Control (MAC) layer, resulting in a minimum of external components and lower application costs.
Because the ZWIR4501 transceiver is based on the IEEE 802.15.4 specification, ZMD highly recommends using the IEEE 802.15.4 specification to complement this user manual. See section 10 for definitions of abbreviations.

1.2 Features

- IEEE 802.15.4 compliant
- Direct Sequence Spread Spectrum (DSSS)
- Burst data rate 40 kbit/s (North America), 20 kbit/s (EU)
- Transmit range 100+ m (LoS)
- Low sleep current = multi-year battery life
- SPI and parallel interfaces
- Thin MAC on PHY = ease of integration, low overall system costs
- Internal transmit/receive switch = external power amplifier capability
- Variable Tx output power = adaptive power density
- Direct conversion radio = reduced cost
- Microcontroller independent = flexibility
- Fractional-N PLL = software configurable frequency control

Integrated MAC Functions

- 128 byte TxFIFO, 256 byte RxFIFO
- CRC generation and checking
- Transmit modes: unslotted and slotted CSMA, GTS; direct Tx
- Frame forming
- Automatic acknowledge generation
- Scan modes: active, passive, energy detection, orphan
- Beacon tracking
- Automatic beacon generation
- Frame filtering
- Timer: TotalTime, Superframe, RxDefer, IFS, Ack, WaitForAck, MaxFrameResponseTime, Scan
- Power saving modes: Global Power Down, Sleep modes while maintaining network time

1.3 RF Frontend Description

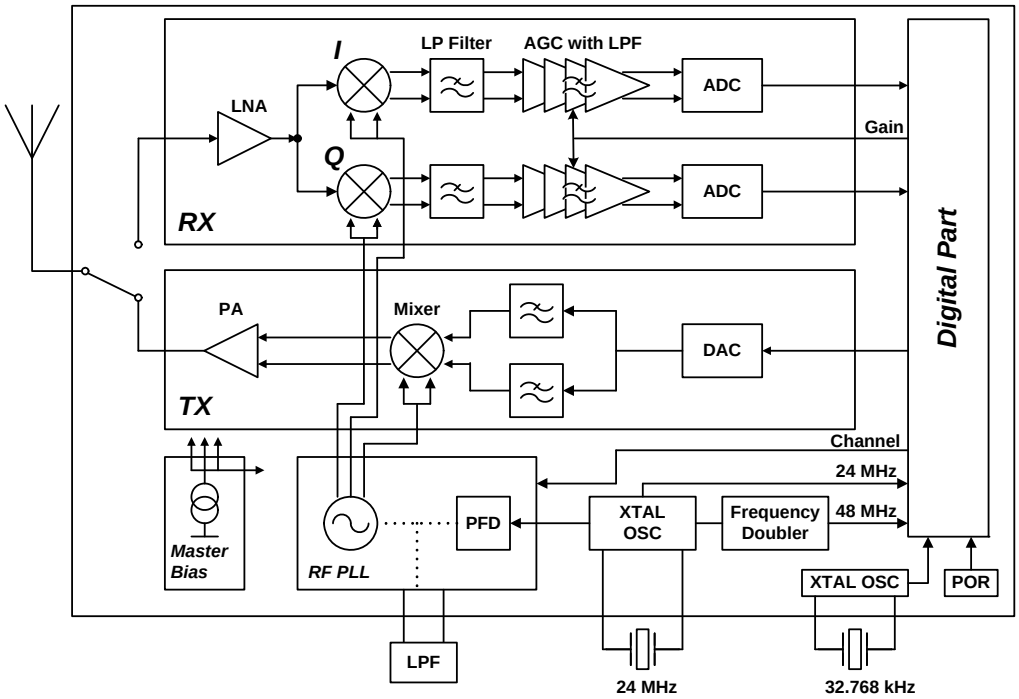


Figure 1.1: Integrated Analog PHY Layer Block Diagram

Receiver Chain

The receiver of the ZWIR4501 uses a direct-conversion architecture (Zero-IF architecture).
The receiver path consists of a 900 MHz low-noise amplifier (LNA) and a mixer, followed by the analog baseband. It contains multi-stage programmable gain amplifiers, low-pass filter sections and analog-to-digital converters (ADC). All remaining functions are carried out in the digital domain, including synchronization, de-spreading, demodulation and the AGC loop control. To extend the dynamic range further, the LNA and mixer gain can be adjusted in the AGC loop.

In normal operation mode, the user or the MAC initiates the reception using the default register values. All control signals (timing, power down) are set automatically.

Transmitter Chain

A direct-conversion architecture is used for the transmitter of the ZWIR4501. The design is fully differential. Only the power amplifier (PA) output is single-ended. No external balun is required.
In normal operation mode, the user or the MAC starts the transmission using the default register values. All control signals (timing, power-down) are set automatically.

Optionally two default register settings of the transmitter can be changed by writing to the *Transmitter Mode Register* (*RTXM*). By default, the internal PA drives 0 dBm (1 mW) to a 50 Ohm off-chip load. This output power can be changed by register settings between 0dBm and -26dBm.

1.4 Pin Assignment

Figure 1.2 shows the pin assignment for the ZWIR4501.

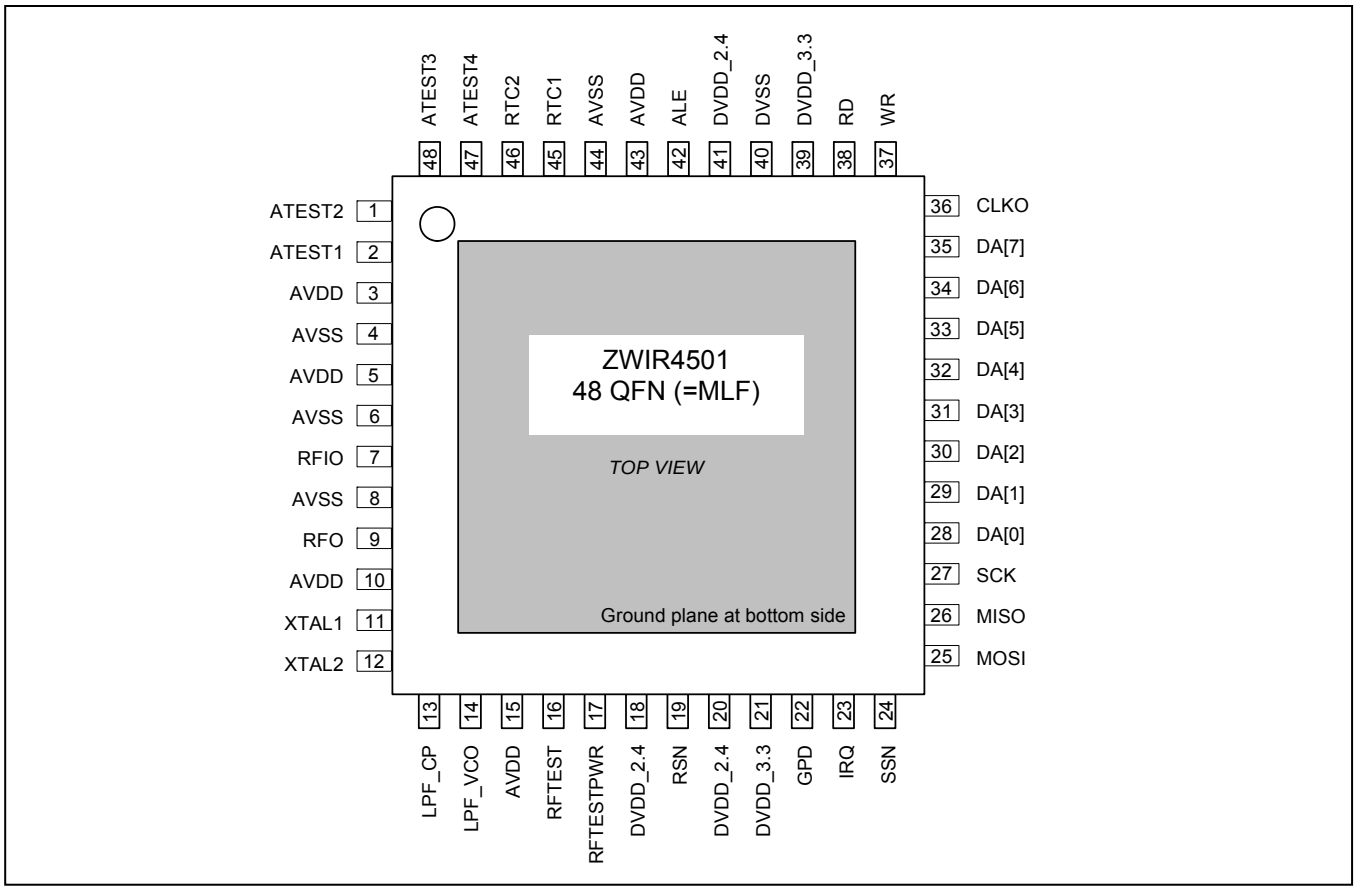


Figure 1.2: Pin Assignment

1.5 Pin Description

Table 1.1 provides a description of the respective pins.

Pin No.	Pin Name	Pin Type	Pin Description
1	ATEST2	Analog I/O	Analog test pin 2, no connection
2	ATEST1	Analog I/O	Analog test pin 1, no connection
3	AVDD	AVDD	Analog power supply, typical 2.4 V
4	AVSS	Ground	Analog ground
5	AVDD	AVDD	RF power supply, typical 2.4 V
6	AVSS	Ground	RF ground
7	RFIO	RF IO	RF receiver input and transmitter output
8	AVSS	Ground	RF ground
9	RFO	RF output	RF transmitter output
10	AVDD	AVDD	Analog PLL power supply, typical 2.4 V
11	XTAL1	Analog input	24 MHz crystal oscillator input

Pin No.	Pin Name	Pin Type	Pin Description
12	XTAL2	Analog output	24 MHz crystal oscillator output
13	LPF_CP	Analog output	Loop filter, charge-pump node
14	LPF_VCO	Analog input	Loop filter, VCO tune node
15	AVDD	AVDD	Analog PLL VCO power supply, typical 2.4 V
16	RFTEST	-	RF test pin, ground
17	RFTESTPWR	AVDD	RF test power supply, typical 2.4 V
18	DVDD_2.4	DVDD	Digital PLL power supply, typical 2.4 V
19	RSN	CMOS input	Chip reset, active low
20	DVDD_2.4	DVDD	Digital core 2.4V power supply (core and pre-driver)
21	DVDD_3.3	DVDD_3.3	Digital IO 3.3V power supply (post-driver)
22	GPD	CMOS input	Global Power Down (from external device), active high
23	IRQ	CMOS output	Interrupt request (to external device), active low
24	SSN	CMOS IO	SPI – slave select not, active low
25	MOSI	CMOS IO	SPI - master out, slave in (in slave mode)
26	MISO	CMOS IO	SPI - master in, slave out (in slave mode)
27	SCK	CMOS IO	SPI - serial clock
28	DA[0]	CMOS IO	Data address, used for parallel interface
29	DA[1]	CMOS IO	Data address, used for parallel interface
30	DA[2]	CMOS IO	Data address, used for parallel interface
31	DA[3]	CMOS IO	Data address, used for parallel interface
32	DA[4]	CMOS IO	Data address, used for parallel interface
33	DA[5]	CMOS IO	Data address, used for parallel interface
34	DA[6]	CMOS IO	Data address, used for parallel interface
35	DA[7]	CMOS IO	Data address, used for parallel interface
36	CLKO	CMOS output	Clock (to external device)
37	WR	CMOS input	Write data address, used for parallel interface
38	RD	CMOS input	Read data address, used for parallel interface
39	DVDD_3.3	DVDD_3.3	Digital IO power supply (post-driver), typ. 3.3 V
40	DVSS	Ground	Digital ground
41	DVDD_2.4	DVDD	Digital core power supply (core and pre-driver), typical 2.4 V
42	ALE	CMOS input	Address latch enable, used for parallel interface
43	AVDD	AVDD	Analog power supply, typical 2.4 V
44	AVSS	Ground	Analog ground
45	RTC1	Analog input	32.768 kHz crystal oscillator input
46	RTC2	Analog output	32.768 kHz crystal oscillator output
47	ATEST4	Analog I/O	Analog test pin 4, no connection
48	ATEST3	Analog I/O	Analog test pin 3, no connection
		Ground	Ground plane at bottom side

Table 1.1: Pin Descriptions

1.6 System Performance Summary

Parameter	Value
System Specifications	
Standard Basis	IEEE Std. 802.15.4™-2003
Spreading Technique	Direct Sequence Spread Spectrum (DSSS)
Modulation Type	Binary Phase Shift Keying (BPSK)
Data Rate Burst	20 kbit/s (EU) and 40 kbit/s (North America, US)
PN Code	15-chip m-sequence
Chip Rate	300 kbit/s (EU) and 600 kbit/s (US)
RF Bandwidth	600 kHz (EU) and 1200 kHz (US)
RF Channel Spacing	2 MHz
Overall Crystal Accuracy	±40 ppm
Architecture	
Receiver (Rx)	Direct down-conversion
Transmitter (Tx)	Direct up-conversion
Phase Locked Loop (PLL)	Sigma-delta fractional-N
Block Specifications	
RF_PLL Frequency Resolution	732 Hz
Tx Output Power	0 dBm (to 50 Ω)
Tx Spurious Emissions	ETSI (EN 300 220) and FCC (Part 15) compliant
Rx Sensitivity	-98 dBm @ channel 0 and -95 dBm @ channels 1-10; PER < 1%
Rx Maximum Usable Input Level	-20 dBm
Rx Selectivity/Blocking Performance	IEEE 802.15.4 compliant + ETSI Rx Class 2
Operational Specifications	
Supply Voltage, Core	+2.2 V to +2.7 V (typical +2.4 V)
Supply Voltage, Digital IO	+3.0 V to +3.6 V (typical +3.3 V)
Temperature Range	-40°C to +85°C
Frequency of Operation	863 MHz to 870 MHz (EU) and 902 MHz to 928 MHz (North America)
Typical Supply Current (Tx)	28 mA
Typical Supply Current (Rx normal, frame reception)	27 mA
Typical Supply Current (Off Mode)	1.3 µA
General Parameters	
Package	48-pin QFN (=MLF™ MicroLeadFrame)
ESD Protection	>2 kV (Human Body Model – HBM)
Interface	SPI and Parallel
External Components	24 MHz and 32.768 kHz XTAL, PLL loop filter (RC), antenna, microcontroller
Process Technology	0.25 µm CMOS

Table 1.2: System Performance Summary

2 Device Specification

Unless otherwise noted, typical values for electrical characteristics over the full range of operating conditions are as follows:
AVDD, DVDD_2.4 = 2.4 V; DVDD_3.3 = 3.3 V; T_a = 27°C

2.1 Recommended Operating Conditions

Parameter	Symbol	Min	Typ	Max	Unit	Notes
Analog supply voltage	AVDD	2.2	2.4	2.7	V	
Digital supply voltage, core	DVDD_2.4	2.2	2.4	2.7	V	
Digital IO supply voltage	DVDD_3.3	3.0	3.3	3.6	V	
Ambient temperature	T_a	-40	+27	+85	°C	Industrial range
Frequency of operation	f_{op}	860		930	MHz	868.3 MHz (EU), 906 MHz to 924 MHz (US)

2.2 Absolute Maximum Ratings

Caution: Operation beyond these values may cause permanent damage to the device or reduce its reliability.
Note: Values are over free-air temperature unless otherwise noted.

Parameter	Symbol	Min	Typ	Max	Unit	Notes
Analog supply voltage	AVDD	-	-	3.5	V	
Digital supply voltage	DVDD_2.4	-	-	3.5	V	
Digital IO supply voltage	DVDD_3.3	-	-	4.6	V	
Input voltage	V_i	-	-	6	V	At CMOS IO
Output voltage	V_o	-	-	4.6	V	At CMOS IO
Analog input voltage	V_{ana}	-	-	3.5	V	At analog IO
Input RF level	P_{in}	-	-	16	dBm	
Storage temperature	T_{strg}	-65	-	150	°C	
ESD protection	V_{esd}	2	-	-	kV	HBM* (100 pF, 1.5 kΩ)

* Human Body Model

2.3 Digital I/O

Module	Symbol	Min	Typ	Max	Unit	Notes
CMOS input	V_{IL}	-0.3		0.8	V	
	V_{IH}	2		5.5	V	
CMOS output	V_{OL}			0.4	V	
	V_{OH}	2.4			V	

2.4 Operation Modes and Current Consumption

Note: Values are for supply current on 2.4 V power supply.

Operating mode	Core active 24 MHz osc.	RTC osc. 32 kHz	RTC logic	Registers content maintained	Analog front-end	Typical current EU mode	Typical current US mode	Unit
Off mode (needs reset)						1.3	1.3	µA
Global Power Down mode (via GPD pin)		X		X		2.2	2.2	µA
Sleep mode (timer-controlled)		X	X	X		2.3	2.3	µA
Idle mode	X	X	X	X		1.25	1.6	mA
Receive mode, frame reception	X	X	X	X	X	27	27	mA
Receive mode, synchronization	X	X	X	X	X	29	30	mA
Transmit mode @ 0 dBm	X	X	X	X	X	28	28	mA

X : active or maintained

Please refer to section *4.11 Operation Mode Description* for further information.

2.5 Startup Time

Parameter	Typ. time	Unit	Notes
Power on to Idle mode	3.0	ms	(2)
Idle mode to Tx (PHY)	180	µs	From end of Tx start command to trailing edge of transmitted frame
Idle mode to Rx (PHY) EU mode US mode	 -100 50	µs	From end of Rx start command to trailing edge of receive frame (1)
Receiver to transmitter turnaround (MAC) EU mode US mode	 600 300	µs	As defined by [1]
Transmitter to receiver turnaround (MAC) EU mode US ode	 600 300	µs	As defined by [1]

Notes:
(1) The ZWIR4501 Rx synchronization algorithm, which starts after PLL and analog Rx power-up, does not require the complete preamble for proper synchronization. Therefore, the frame could start earlier than the start of Rx synchronization, which leads to a negative time in EU mode.

(2) The RTC requires 730 ms stabilization time after a power-up. Please refer to chapter 4 Integrated HW-MAC for further information.

2.6 RF Parameter Summary

Parameter	Min	Typ	Max	Unit	Notes
Transmitter					
Nominal output power P_{out}	-3	0	3	dBm	Output power at 50 Ω
Low out power modes		-16		dBm	LP1
		-20		dBm	LP2
		-26		dBm	LP3
Chip rate		300		kbit/s	@ channel 0
		600		kbit/s	@ channels 1 to 10
Error vector magnitude (EVM)		10		%	As defined by [1]
Receiver					
Sensitivity		-98		dBm	At packet error rate (PER) <1% [1], EU mode
		-95		dBm	At packet error rate (PER) <1% [1], US mode
Maximum usable input power		-20		dBm	
Input-referred IP3		-25		dBm	
Input-referred IP2		25		dBm	
Adjacent channel rejection		30		dB	As defined by [1]
Alternate channel rejection		45		dB	As defined by [1]
PLL					
Frequency range	860		930	MHz	
Crystal reference frequency		24		MHz	
Loop bandwidth		100		kHz	
Frequency resolution		732		Hz	
Channel spacing		2		MHz	For IEEE 802.15.4
Phase noise		-80		dBc/Hz	10 to 100 kHz offset

Check with local authorities for frequency regulations. Contact ZMD support for non IEEE 802.15.4 compliant frequency applications.

3 Application Circuit and External Components

The ZWIR4501 requires few external components, which allows for a small module form factor and low bill of material (BoM) costs. Figure 3.1 shows an example of an application circuit and its required components. Additionally, a separate microcontroller is required, along with its support components. The standard microcontroller interfaces are described in paragraph *3.5* on page 18.

Note: Please refer to ZWIR4501 application notes [7] for reference design details before using the application circuit example for a product design.

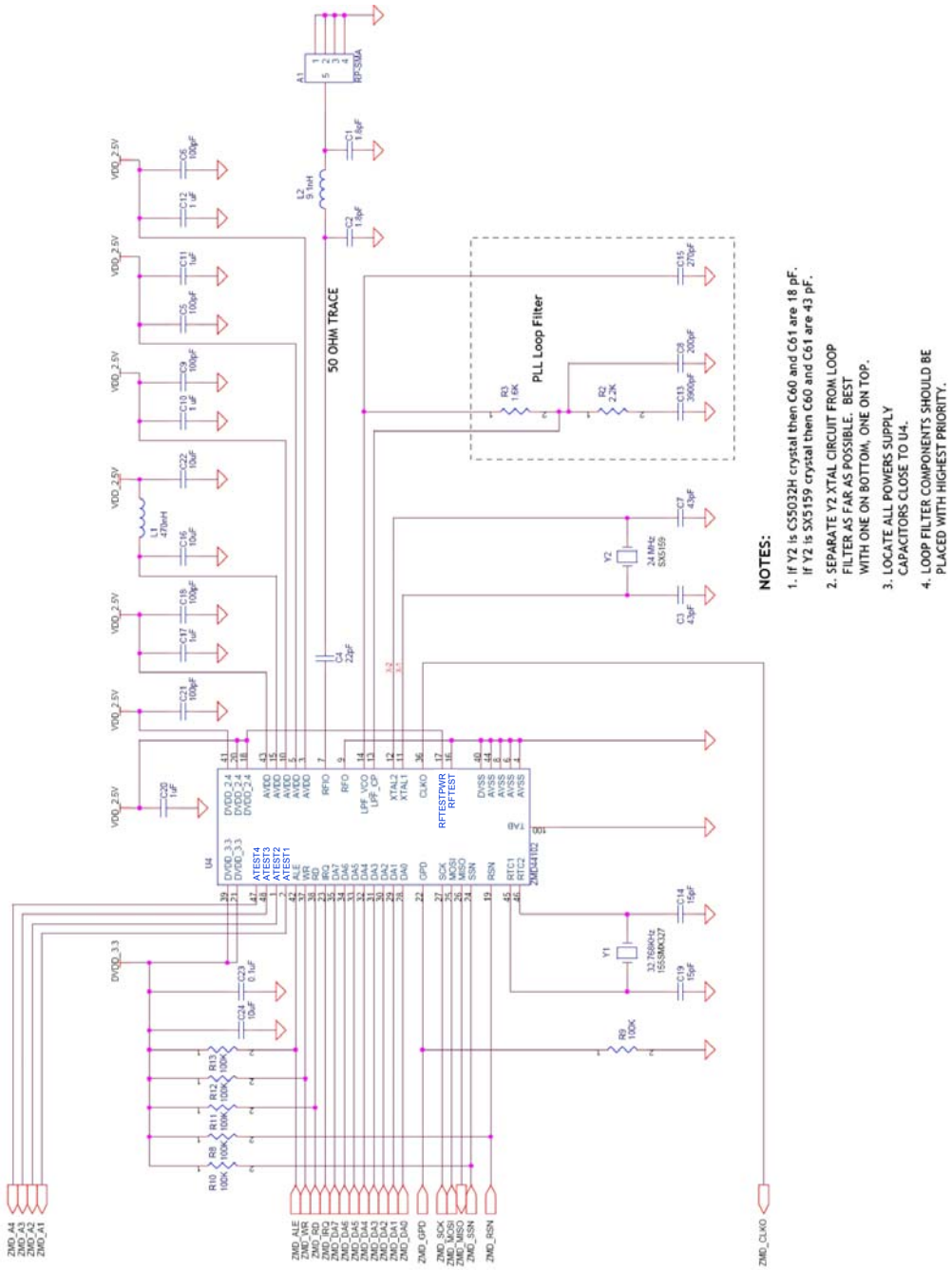

50 OHM TRACE
PLL Loop Filter
NOTES:
1. If Y2 is CS5032H crystal then C60 and C61 are 18 pF.
If Y2 is SX5159 crystal then C60 and C61 are 43 pF.
2. SEPARATE Y2 XTAL CIRCUIT FROM LOOP FILTER AS FAR AS POSSIBLE. BEST WITH ONE ON BOTTOM, ONE ON TOP.
3. LOCATE ALL POWERS SUPPLY CAPACITORS CLOSE TO U4.
4. LOOP FILTER COMPONENTS SHOULD BE PLACED WITH HIGHEST PRIORITY.

Figure 3.1: Application Circuit Example [5]

3.1 Crystal Oscillator

The ZWIR4501 utilizes with two crystal oscillators, one for 24 MHz and one for 32.768 kHz. The 32.768 kHz clock is used as a real time clock during the Sleep mode and for timing adjustments during the beacon-enabled mode. The 24 MHz clock is used for most of the digital blocks and as the reference frequency for the RF PLL.

3.1.1 Reference Crystal Oscillator (24 MHz)

A 2-pin Pierce oscillator with an on-chip biasing resistor provides the necessary reference frequency at 24 MHz. This frequency is used for the digital clock supply, timing calculations and the PLL that generates the RF carrier frequency. For the receive modes, the internal circuitry doubles the reference frequency to achieve the digital processing speed during code acquisition. This oscillator is active only in Idle, transmit and receive power modes (see section 2.4 Operation Modes and Current Consumption for further details).

When the internal oscillator is used, C4 and C5 are required as load capacitors for the parallel resonance crystal. The values C4 and C5 are determined by the specific crystal used. The overall load capacitance is composed of the actual values of C4 and C5 and the parasitic values of the PCB layout and the internal parasitic capacitance of the ZWIR4501, which is 0.65 pF on each pin. For the recommended SMI SX5159 crystal the values for C4 and C5 are typically 43 pF. The user should experiment with values near 43 pF to determine the capacitor value for best frequency accuracy in a production environment. Any deviation on this system part will result in a large deviation on the carrier frequency and decrease system performance.

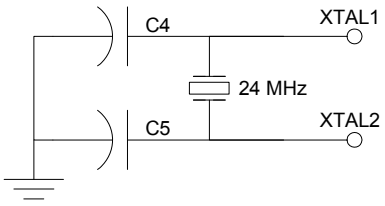


Figure 3.2: 24 MHz Crystal Oscillator – External Components

3.1.2 Low Power Crystal Oscillator (32.768 kHz)

The 32.768 kHz crystal oscillator is designed for extremely low power operation, and it always runs when power is applied to the device. There is also a user programmable option to power down the oscillator (see description on Off mode, section 0). The oscillator provides the time reference for the on-chip real time clock. The oscillator utilizes an amplitude-controlled 2-pin Pierce oscillator with an on-chip biasing resistor. The recommended crystal to be used with this oscillator is the SMI 155M327 because of frequency tolerance and temperature range specifications. The load capacitor values are determined by the specs of the particular crystal selected.

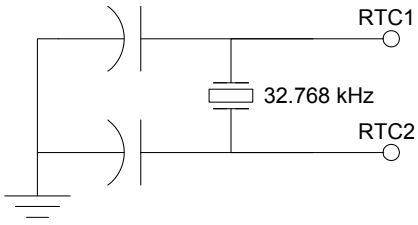


Figure 3.3: 32.768 kHz Crystal Oscillator – External Components

The ZWIR4501 provides external access to the 24 MHz or 32.768 kHz clocks for unique system implementations. Refer to section 3.4 for further information on CLKO.

3.2 RF Phase Locked Loop

A fractional-N phase locked loop (PLL) architecture is used by the ZWIR4501. All functions are integrated on chip except the loop filter. The external loop filter circuitry is depicted in Figure 3.4. The 24 MHz crystal (see section 3.1.1) provides the reference frequency for both EU and US bands.

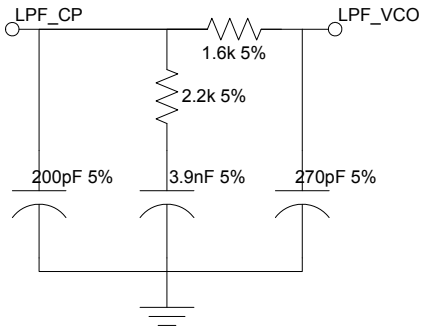


Figure 3.4: PLL-Loop Filter

In normal operation mode, the user sets the frequency channel of the RF PLL before transmission or reception by writing to the phyCurrentChannel register (RPCC). The data rate (EU: 20 kbit/s; US: 40 kbit/s) is adjusted automatically according to the selected channel. The channel numbers are defined by the IEEE 802.15.4 standard [1]. Figure 3.5 illustrates the channel allocation in the 900 MHz band.

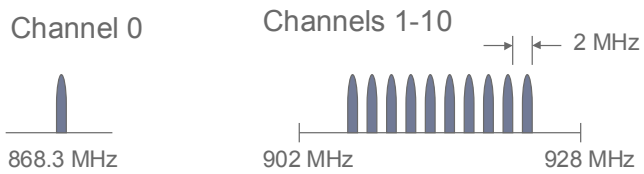


Figure 3.5: Channel Allocation in the 868 and 915 MHz Bands

3.3 Antenna Connection and Power Amplifier Configuration

By default, the receiver input and the transmitter output use the same pin (RFIO). The integrated antenna switch disconnects the respective components in transmit and receive modes. The antenna must be connected via an external capacitor for DC blocking (e.g. 22 pF).

The ZWIR4501 can drive an external power amplifier to increase the coverage. Figure 3.6 depicts a block diagram using the ZWIR4501 with an external power amplifier.

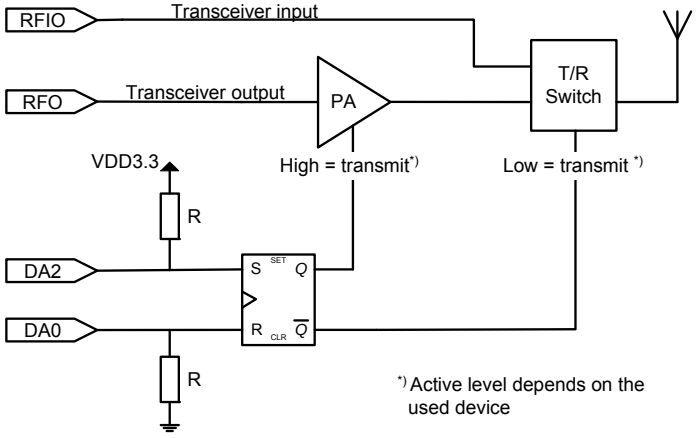


Figure 3.6: Power Amplifier Configuration Block Diagram

The transmitter uses the RFO pin as the RF output and the RFIO as the RF input. The external power amplifier and the external RF switch can be controlled (switching on or off) using a flip-flop and the digital transceiver outputs DA0 and DA2. The digital signals DA0 and DA2 provide short pulses (of about 60 µs duration, high active) that can be used together with a flip-flop to determine when the transceiver is either in transmit mode or in receive mode. The transceiver pins DA0 and DA2 need to be configured to digital output by the following register settings after the startup register setup procedure:

- Write to register address 0x05 the value 0x08
- Write to register address 0x42 the value 0x20
- Write to register address 0x52 the value 0x20
- Write to register address 0x55 the value 0x05

Please see section 5.17 for further information on the recommended startup register setup.

3.4 Clock Output (CLKO)

The ZWIR4501 can provide an external clock signal on its CLKO pin for optional microcontroller timing. The clock output can be configured independently for Sleep/Global Power Down mode and for the normal operating modes (i.e., all modes except Sleep and Global Power Down mode). The configuration is set up in the ClkOutConfig register. By default, the CLKO provides 6 MHz in normal mode and 32.768 kHz in Sleep/Global Power Down mode. To reduce the power consumption, switching off the clock output signal is recommended, if it is not being used by the application. The clock on the CKLO pin can be configured for external microcontroller clock support by the ClkOutConfig register.
During the Global Power Down and Sleep modes, the clock output signal is switched to 32.768 kHz or to selectable fractions of 32.768 kHz for reduced current consumption. During all other states, the 24 MHz clock or selectable fractions of 24 MHz can be used on CLKO.
Pin 36 can directly drive clock input up to 4 mA.

3.5 Interfacing

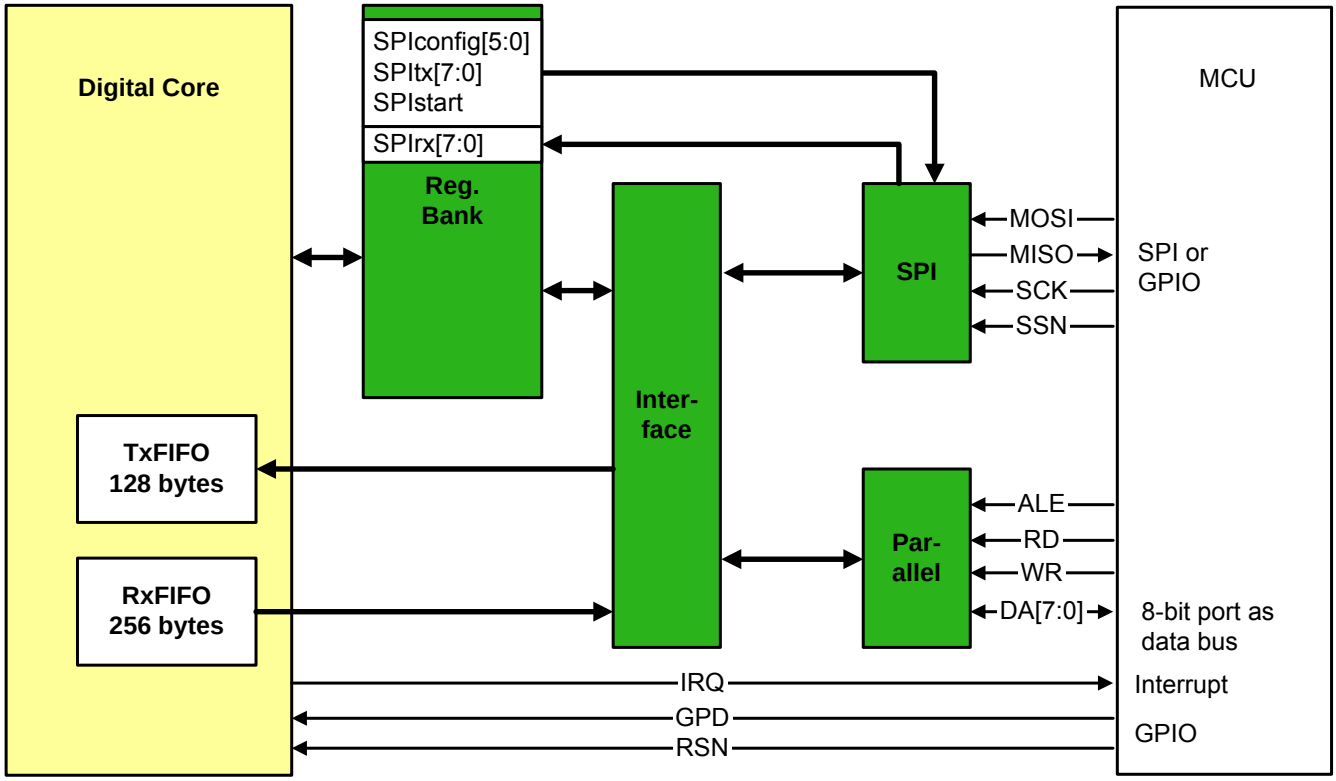


Figure 3.7: Interface Block Diagram

The ZWIR4501 provides two different interfaces: a standard-based SPI (MOSI, MISO, SCK, SSN) and a parallel interface. Both can be used to access the internal register bank, as well as the transmit (TxFIFO) and receive (RxFIFO) FIFOs. Additionally, it has an IRQ output, a dedicated Global Power Down (GPD) input and a reset not (RSN) input. The ZWIR4501's SPI can be connected to an MCU's hardware SPI implementation, or it can be controlled by an MCU's I/O lines. It is recommended that an input port that can be handled by interrupt control be used for the IRQ line.

By default, both the parallel interface and the SPI in slave mode are available. For proper operation, the unused interface must be disabled. The parallel interface is disabled by setting RD, WR, and ALE to high and putting the DataAddress[7:0] bus into High-Z state. The SPI is disabled by setting SSN to high.

The interfaces can be used only if the transceiver's digital core is active. Please see section 2.4 for further information on operating modes.

3.5.1 Power Up

The ZWIR4501's power up sequence and its timing are shown in Figure 3.8.

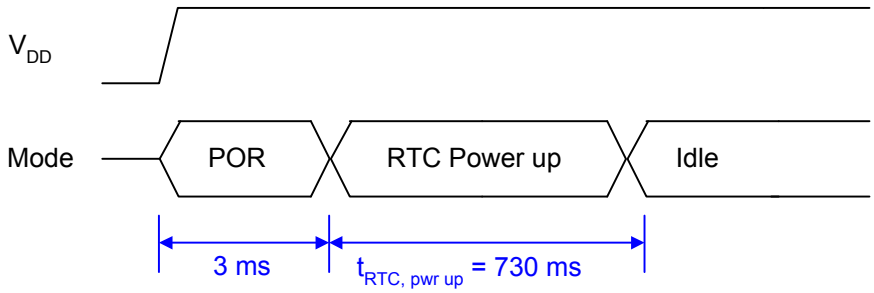


Figure 3.8: Power Up Sequence

For best performance results, set the trim values of the ZWIR4501 after the power-up sequence. Refer to section 5.17 for configuring ZWIR4501's trim values after the power-up sequence. The power-up time of 730 ms is needed to settle the 32 kHz RTC (osc.). After this time the ZWIR4501 is fully functional. The registers can be accessed after POR. For some functionality the RTC is required, e.g. beacon mode and GPD.
Do not supply DVDD_2.4 before DVDD_3.3 to the IC.

3.5.2 Reset (RSN)

The ZWIR4501 can be reset by using the RSN pin (pin #19). This pin is active low, i.e. setting this line to low starts the reset procedure. The reset procedure and its timing are shown in Figure 3.9.

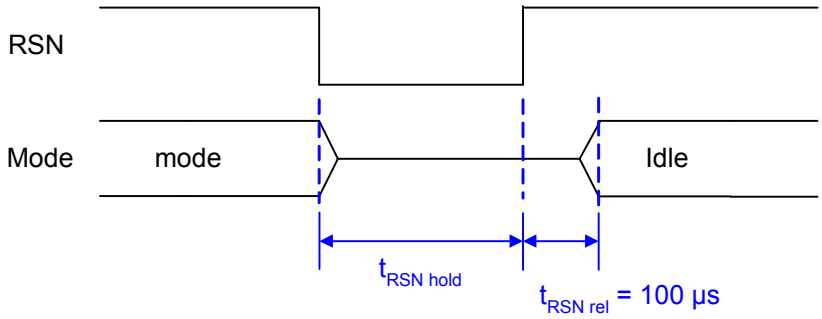


Figure 3.9: Reset Sequence

Table 3.1 gives the values for the RSN hold time t_{RSN} in different operating modes.

Mode	RSN hold time ($t_{RSN hold}$)
Idle, Tx, or Rx mode	$t_{RSN hold 1}$ = 500 µs
Sleep or Global Power Down	$t_{RSN hold 2}$ = 3 ms
Off mode	$t_{RSN hold 3}$ = 730 ms

Table 3.1: Reset Timing

For best performance results set the trim values of the ZWIR4501 after the reset sequence. Refer to section 5.17 for further information on the recommended trim values. See section 0 for further information on low power modes.

3.6 Synchronous Serial Peripheral Interface (SPI)

The SPI is used by default in slave mode. The master mode is not described by this document. If the master mode is desired, please contact ZMD for further information. The SPI is configured via the SPIconfig register. The SPIconfig register can be used to switch the interface to master mode to operate with another slave and to select the clock phase and clock polarity modes. In that case, the parallel microcontroller interface is used to control the ZWIR4501. The SPI provides standard lines, i.e., MISO, MOSI, SCK and SSN.

The ZWIR4501's SPI supports both clock polarity modes (CPOL=0 and CPOL=1) and both clock phase modes (CPHA=0 and CPHA=1). The setting of SPIconfig register determines which mode is used. The default configuration is CPOL=0 and CPHA=0.

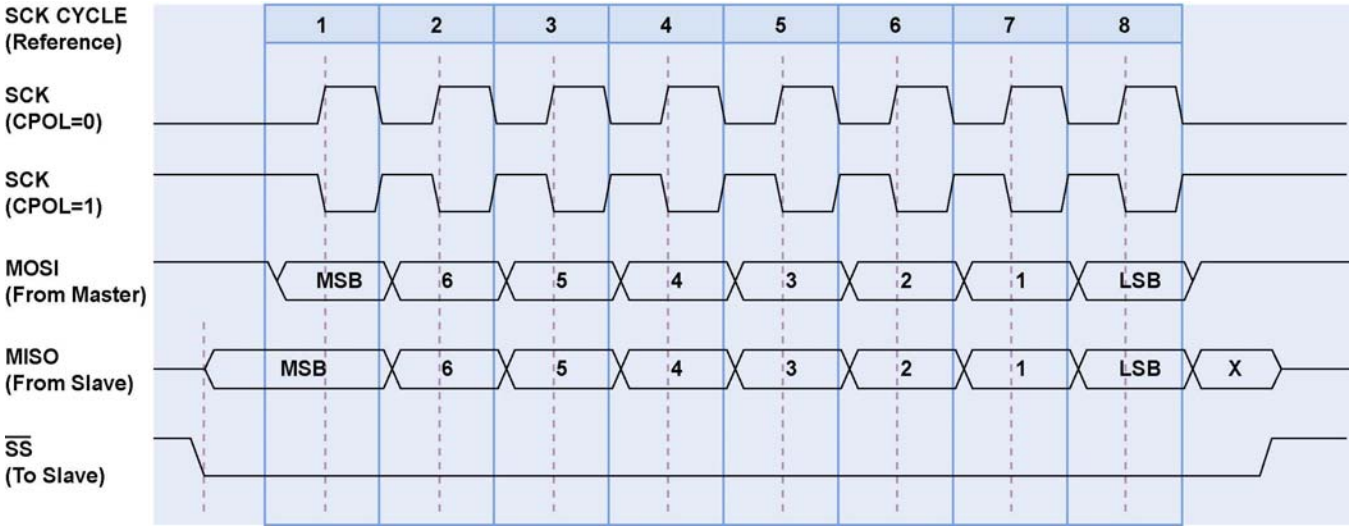


Figure 3.10: CPHA=0, SPI Transfer Format

The ZWIR4501 uses a data transfer protocol allowing single and multiple byte read/write access. All bytes are transmitted using the MSB first and the LSB last. If the SSN line is set to low level, the IC's MISO is configured as an output pin. Otherwise, when SSN line is high, MISO is configured as an input pin. This allows other devices to use the same bus.

The read/write protocol always starts by setting SSN (slave select not) to low when accessing the ZWIR4501 through the SPI and then writing two bytes to the SPI slave via the MOSI line. The MSB of the first byte is the read/write indicator. A high bit indicates read access and a low bit indicates write access. The read/write bit is followed by the length[6:0] descriptor N. It controls the length of the data frame D_0[7:0] to D_{N-1}[7:0]. N must be in the range 1 to 127. The second byte is the address[7:0].
For TxFIFO and RxFIFO access, the addresses are 0x80 and 0x81 respectively. Note that the TxFIFO allows only write access and the RxFIFO allows read and write access (please refer to the register description of the RxFIFO). During register bank access, a number of N bytes is read starting from address[7:0] up to address[7:0]+(N-1). If the FIFO locations 0x80 or 0x81 are within this range, they are skipped and the read/write access is continued at location 0x82.

For write access, the address[7:0] byte is followed by the data frame D_0[7:0] to D_{N-1}[7:0]. The slave select not (SSN) line is low during the complete write transfer. However, SSN high gaps can be inserted between each byte.
In the read access protocol, the data frame is shifted out by the slave on the MISO line. Before each data byte, an SSN high gap is required. As with write access, an SSN high gap can be inserted before the address[7:0] byte.

Note that the SPI cannot be used in GPD or Sleep mode. If the SPI protocol has been corrupted or violated, a reset of the ZWIR4501 is required.

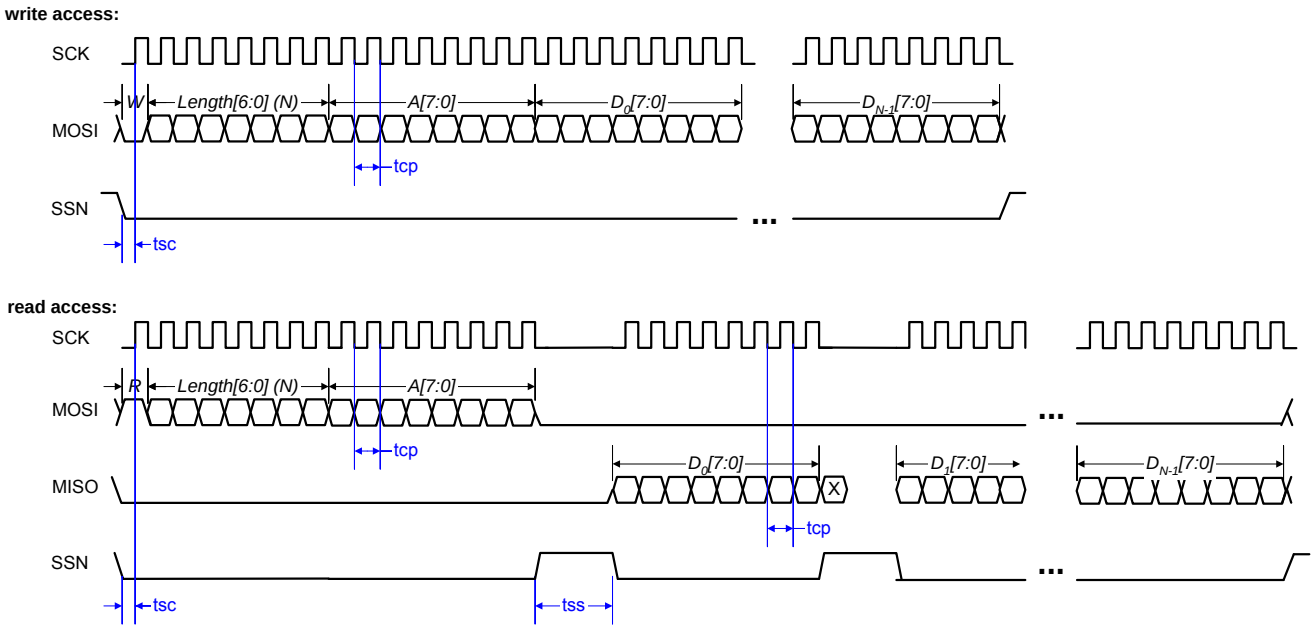


Figure 3-2: Transfer Protocol (SPI Slave Mode), CPHA=0 CPOL=0

The timing parameters are listed in the following table: Table 3.2: SPI Timing parameters.

Parameter	Description	US Mode		EU Mode	
		Min	Max	Min	Max
tcp	SCK clock period	0.50 µs		1.00 µs	
tsc	SSN low to first active SCK edge	0.25 µs		0.50 µs	
tss	SSN high pulse width	1.00 µs		2.00 µs	

Table 3.2: SPI Timing parameters

3.7 Parallel Interface

The parallel interface consists of the bi-directional DataAddress[7:0] bus and the control inputs read (RD), write (WR) and address latch enable (ALE). The direction of the DA[7:0] bus is controlled by the RD input.
If RD is high, DA[7:0] pins are in input mode. Setting RD sets DA[7:0] with the delay of trvd to output mode.
The timing diagram for read and write access is listed in the following table: Table 3.3: Parallel Interface Timing Parameters.

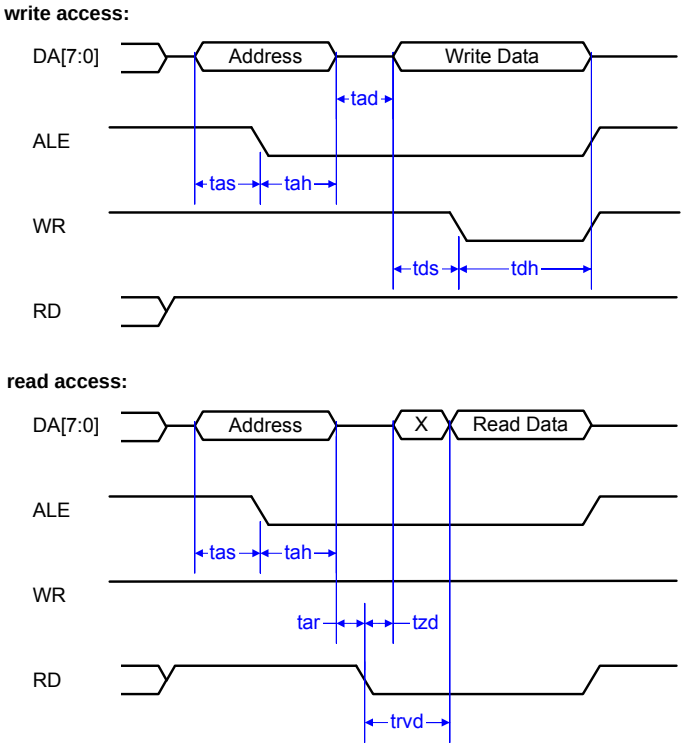


Figure 3.11: Parallel Interface Write/Read Access

Parameter	Description	US Mode		EU Mode	
		Min	Max	Min	Max
tas	Address setup time	0		0	
tah	Address hold time	200 ns		200 ns	
tad	Address to data time	0		0	
tds	Data setup time	0		0	
tdh	Data hold time	0.5 µs		1 µs	
tar	Address to RD low time	0		0	
tzd	High-Z to data time	0	10 ns	0	10 ns
trvd	Read low to valid data		0.5 µs		1 µs

Table 3.3: Parallel Interface Timing Parameters

4 Integrated HW-MAC

4.1 Overview

The ZWIR4501 is an 868/915 MHz IEEE 802.15.4 compliant transceiver. It integrates the complete PHY layer including the RF front-end and the digital signal processing with hardware support for the MAC layer and an interface to an external microcontroller.

The ZWIR4501 can be controlled via a SPI or a parallel interface. It also provides a dedicated IRQ output, a Global Power Down (GPD) input and a chip reset (RSN, active low) input.

The hardware MAC (HW-MAC) contains a 128 byte transmit FIFO (TxFIFO), a 256-byte receive FIFO (RxFIFO), the frame composition and decomposition, automatic acknowledge generation, the CRC generation and check, and a MAC controller together with several support timers.

The MAC controller provides different operating modes:

- Transmitting (slotted and unslotted CSMA, GTS; direct)
- Receiving
- Scan (ED, Passive, Active, Orphan)
- Beacon Tracking
- Automatic Beacon Generation
- Automatic spanning of slotted CSMA and slotted “wait for frame response” procedure over multiple contention access periods (CAP)

In order to run the implemented operating modes with a minimum of software support and a reduction in the interrupt requirements and workload from the microcontroller, a set of timers is integrated to support the HW-MAC:

- TotalTime
- Superframe
- RxDefer
- IFS, Ack
- WaitForAck
- MaxFrameResponseTime
- Scan
- General Purpose

The ZWIR4501 uses two different clocks: a 24 MHz system clock (for the digital core) and a 32.768 kHz real time clock (RTC). It supports different sleep and power down modes. During the Sleep and Global Power Down modes, the 24 MHz oscillator is shut down to reduce the power consumption. The RTC is running during the Global Power Down and Sleep modes, maintaining the network time. This enables the ZWIR4501 to go to Sleep mode at any time within a superframe without losing track of the superframe time, if operating as a slave device.

4.2 Block Diagram

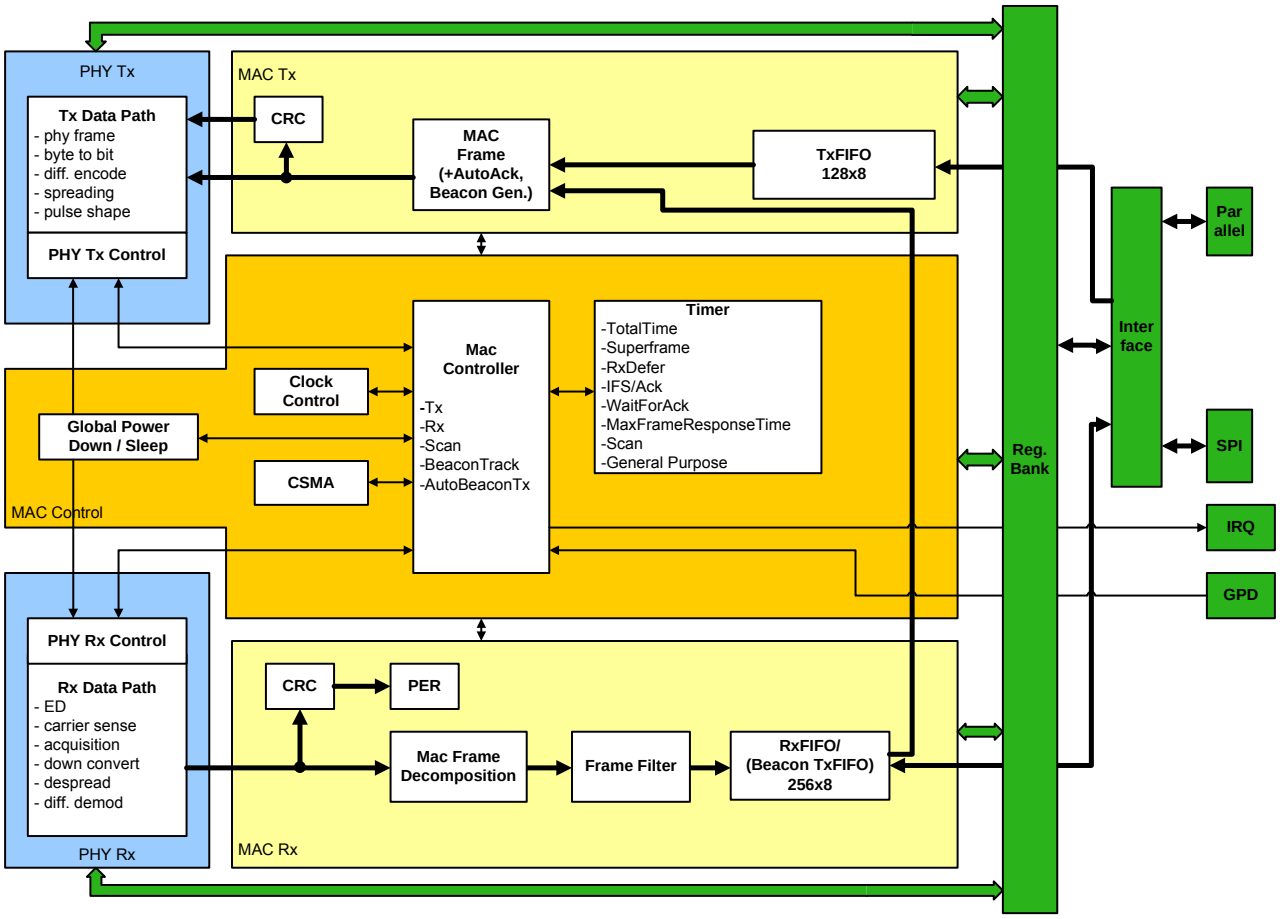


Figure 4.1: Digital Core Block Diagram

4.3 MAC Control

The HW-MAC layer is controlled via the macControl register by a set of defined commands, most of which are operating-mode-dependent. The HW-MAC also uses the macControl register to provide feedback to the software about command acceptance status. After a command release, the MCU can read this information and use it for debugging purposes or exception error-handling routines.

There are three possible scenarios shown in *Figure 4.2*. In case A, a valid command has been written to macControl. Hardware successfully executes the command and resets the macControl register to 0x00. In case B, a command that is undefined or invalid in the current operation mode is requested. HW-MAC responds with a command error, setting macControl to 0x1F. A command error is also indicated by an interrupt with the IRQreason CmdError and the macTimerControlStatus CmdError/CmdErrorT.
In case C, a command is valid but cannot be executed immediately because of the internal HW-MAC status. An example is a macControl command TxOn during an ongoing data reception. The command is queued until the reception is complete and then executed. The execution is indicated by the 0x00 value.
Only one command can be buffered while the internal HW-MAC is executing the previous command.

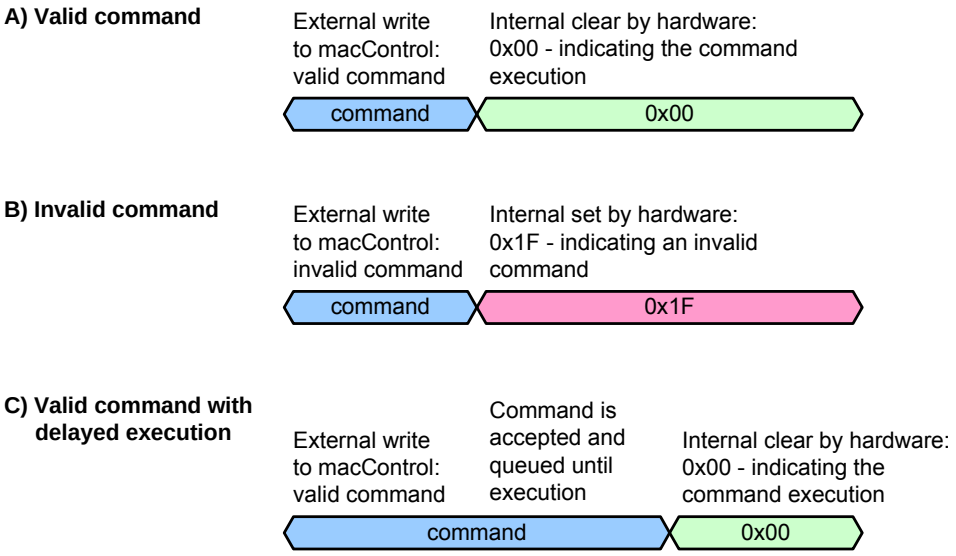

A) Valid command
External write to macControl: valid command
Internal clear by hardware: 0x00 - indicating the command execution
command
0x00
B) Invalid command
External write to macControl: invalid command
Internal set by hardware: 0x1F - indicating an invalid command
command
0x1F
C) Valid command with delayed execution
External write to macControl: valid command
Command is accepted and queued until execution
Internal clear by hardware: 0x00 - indicating the command execution
command
0x00

Figure 4.2: Scenarios for macControl Register

4.4 Interrupts

The ZWIR4501 has 23 mask-able interrupt conditions, which are grouped by eight different interrupt reasons. Most of the interrupt conditions correspond to status updates of the HW-MAC. The IRQ processing is shown in Figure 4.3 below. If there is no interrupt, the IRQ line is high. If an unmasked interrupt condition occurs, the IRQ output is set to low. On each of the following unmasked interrupt conditions, the IRQreason register is updated. In response to an IRQ, software reads the IRQreason register and the associated status register. The IRQreason register is automatically cleared to 0x00 after the register has been read. This will reset the IRQ line to high as well. If another interrupt condition occurs only on the read access to the IRQreason register, the IRQ line will stay low, indicating that an interrupt is still pending. In this case, the IRQreason register is overwritten with the new interrupt reason. After an interrupt is issued the corresponding status register is updated to provide more information on the actual reason for the issued interrupt.

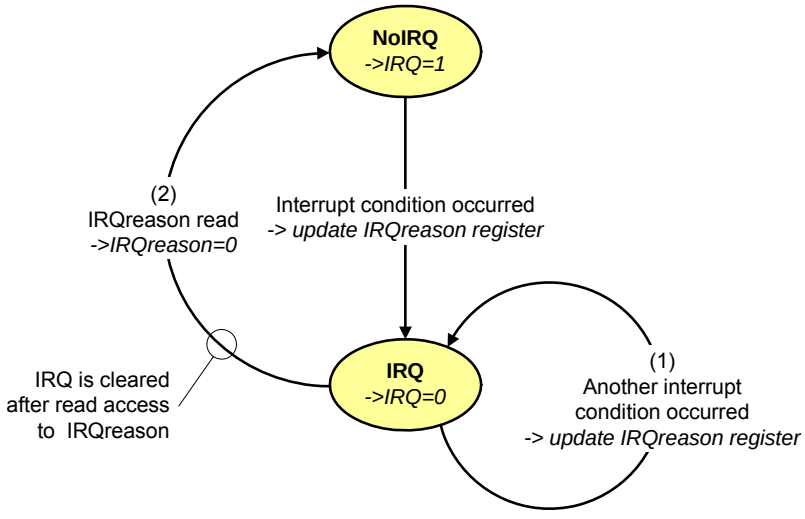


Figure 4.3: Interrupt Processing

See Section *5.2* for further information on interrupts and their registers.

4.5 Frame Handling

The ZWIR4501 provides a 128 byte TxFIFO and 256 byte RxFIFO. Both FIFOs can be flushed independently at any time by the macControl commands TxFifoFlush and RxFifoFlush.

The TxFIFO can store the content for a minimum of one transmit frame. It can be used in two different modes which are selected by the DirectFifoAccess bit in the macTxConfig register.
In the default mode, the TxFiFO is used, together with the some register bank space to store the transmit frame content (MSDU). In this mode the sequence number, frame control information, source and destination addresses and MSDU length indicator are stored in the register bank. Only the MSDU is written to the TxFIFO. The transmit frame is then constructed by an integrated frame former, which also composes the address field according to the addressing modes of the frame control field (refer to section 4.11.2.2).

Setting the DirectFifoAccess bit in the macTxConfig register enables the DirectFifoAccess mode. In the DirectFifoAccess access mode, the complete MAC frame, including the frame control bytes, sequence number, address bytes and MSDU, are written to the TxFIFO. A length byte indicating the length of the complete MAC frame must be written into the TxFIFO before the MAC frame. The length value must include the whole payload length, including MAC header and frame payload length. The CRC check sum is added automatically to the sent frame. The two CRC bytes should not be included in the length value. In the DirectFifoAccess mode, the integrated address field composition is bypassed.
During DirectFifoAccess mode, the ZWIR4501 uses the Acknowledge requested bit in the mhrFc1Tx register to determine if an inbound acknowledgement frame is expected. Note that the ZWIR4501 does not use the Acknowledge requested bit of the transmit frame control field in the TxFIFO to make this determination. If the Acknowledge requested bit in the mhrFc1Tx register is not set, and a data frame is transmitted using the macControl command TX_ON, an Ack timeout interrupt is not generated. If the Ack timeout notification is desired, the Acknowledge requested bit in the mhrFc1Tx must be set prior to issuing the macControl command TX_ON. Note that, in DirectFifoAccess mode the IFS following the transmit frame should be calculated by SW and both the T_SIFS and the T_LIFS register should be temporarily overwritten by the calculated IFS.

The RxFIFO can be used alternatively as TxFIFO for beacon transmissions. In beacon transmission mode, a TxFIFO usage conflict can occur if the TxFIFO is still occupied with data from a pending/ongoing transmission and if the MCU needs to write the MSDU of the next beacon into the TxFIFO. To avoid this problem, use the RxFIFO as an alternative for beacon generation. This mode is enabled by setting the UseRxFIFO bit in the macBcTxConfig register. In this case, the RxFIFO is used in the same way that the TxFIFO is used in direct TxFIFO access mode, and the complete beacon frame content must to be written to the RxFIFO. In this case the IFS following the beacon frame should be calculated by SW and both the T_SIFS and the T_LIFS register should be temporarily overwritten by the calculated IFS.

For auto acknowledge generation, the acknowledge frame is generated automatically, with the sequence number of the previous received frame and the frame type set to Ack (0b010). The remaining frame control field bits [15:3] can be adjusted by the firmware in the register mhrAckFc1Tx and mhrAckFc2Tx. As a rule, the frame pending bit mhrAckFc1Tx[1] must be controlled by the MCU. For forward compatibility reasons, the other bits can also be modified.
For received frames, the MAC header (MHR) and payload (MSDU) are queued in the 256 byte RxFIFO. Each frame is deposited in the RxFIFO, starting with a length indicator followed by the MHR and the MSDU. The length indicator defines the length of the MHR and the MSDU. A two byte link quality indicator value (LQI, refer to section 4.6 for further information) is appended (behind the MSDU) to each stored packet (low byte first). This is enabled by default and can be disabled by the FifoStoreLQI bit in the macRxConfig register. The MHR and the CRC field of the last received frame are also held in the register bank. The sequence number mhrSquNbRx is used for the acknowledgement frame in the case of auto acknowledge generation. The storage of acknowledge frames can be disabled by the FifoStoreAck bit in the macRxConfig register.

Frames not passing the CRC or frame filter check are removed from the RxFIFO at the end of the reception.

The macFramePend register indicates the number of frames in the RxFIFO. It is incremented after each stored frame. Software can read the macFramePend each time a frame is read from the RxFIFO. The macFramePend

content will be decremented by 1 automatically after each read access. In the case that a receive interrupt is generated and the associated macRxStatus is 0x00 or 0x01, the macFramePend register can be used to check for a received frame.

FIFO overflow and underflow conditions are reported via an interrupt with the IRQreason FIFO. The appropriate FIFO status is monitored in the macFifoStatus register. For both reading from an empty FIFO and writing to a full FIFO, the access is blocked by the FIFO control to prevent the FIFO content from over-writing or over-reading.

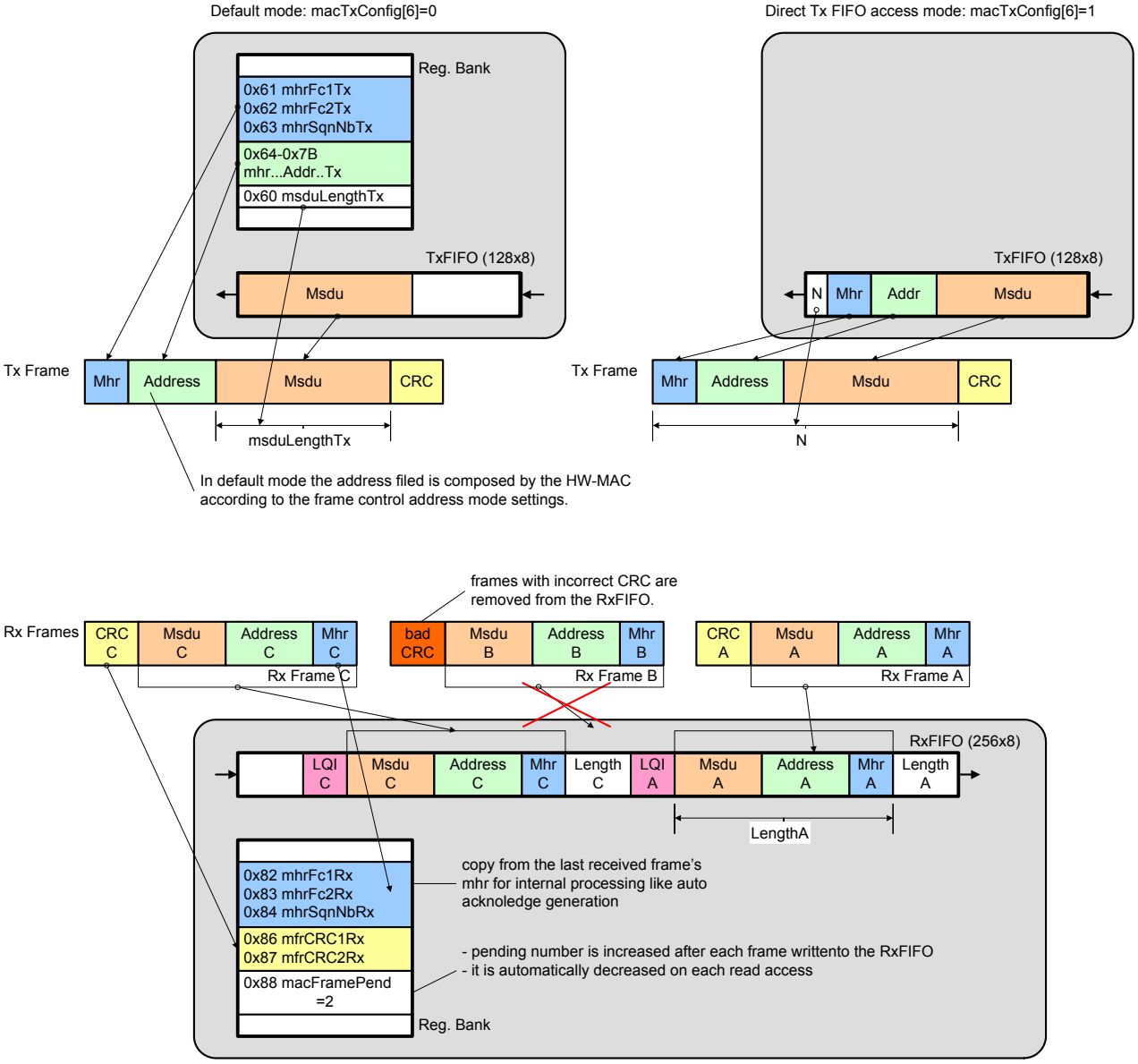


Figure 4.4: Tx/Rx Frame Handling

4.6 Link Quality Indicator (LQI)

The IEEE 802.15.4 standard defines a Link Quality Indicator (LQI, see [1] section 6.7.8). This value can be estimated by using the value of the AgcLvl register after a successfully received frame (before the receiver is switched on again). It is not recommended to use the two bytes that are stored after a received frame to the RxFIFO to estimate the LQI.

The mapping of the AgcLvl register value to the LQI value is shown by Figure 4.5.

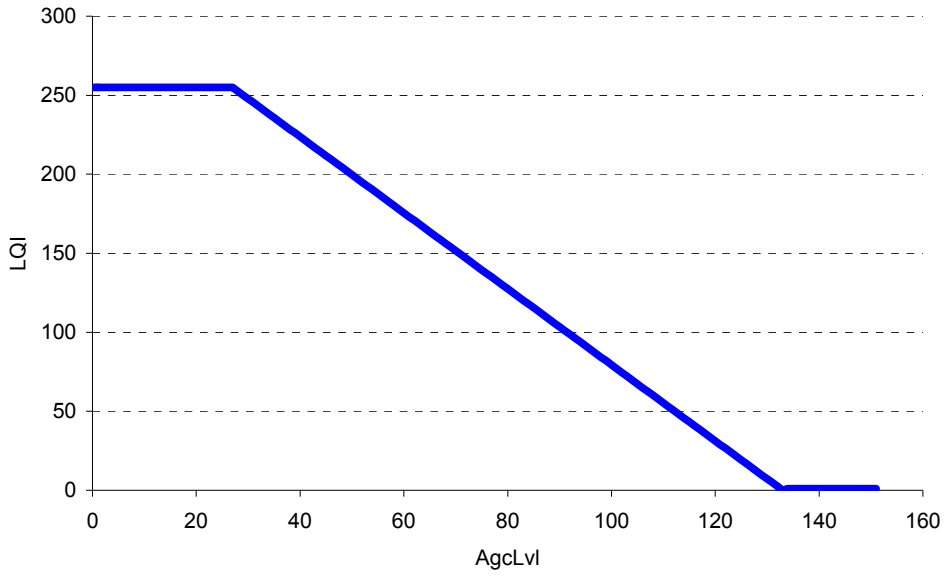


Figure 4.5: LQI over AgcLvl

The LQI value can be calculated by the following simplified equation:

LQI = Integer (320 - 2.4 * AgcLvl) , valid for: $27 \leq AgcLvl \leq 133$

LQI = 0, valid for: AgcLvl > 133

LQI = 255, valid for AgcLvl < 27

4.7 Receive Signal Strength Indicator (RSSI)

Over a certain dynamic range, the measured AgcLvl value is an indirect proportional input signal level. The AgcLvl can be used to estimate the receive signal strength of the last successfully received frame. The graph in Figure 4.6 shows the correlation between the AgcLvl and the input power level.

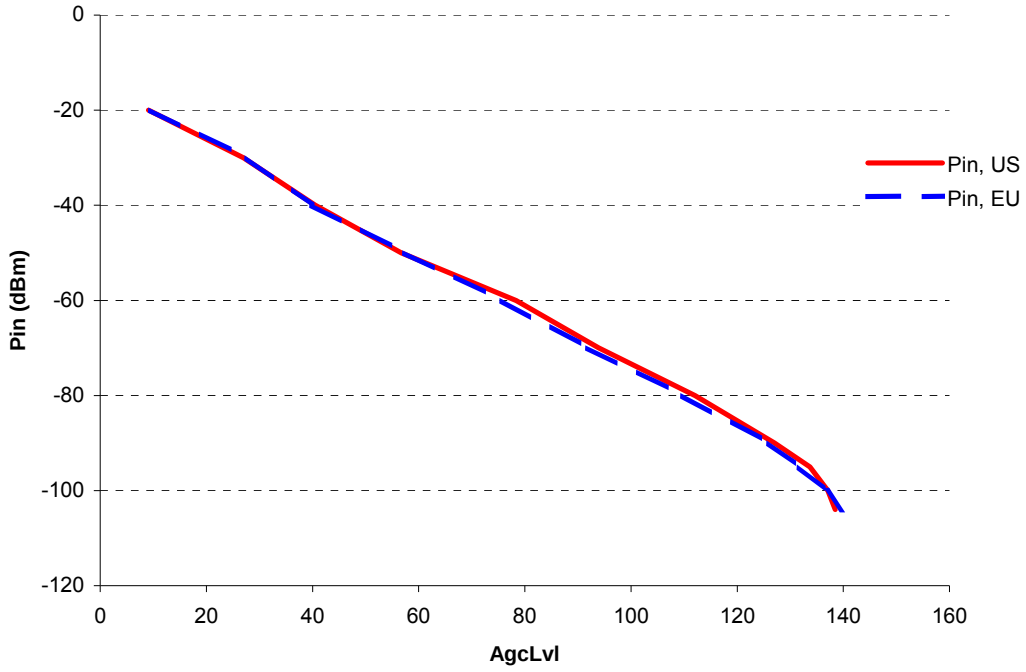


Figure 4.6: Typical AgcLvl vs. RF Input Power (Pin)

For input levels below -95 dBm, the estimation of the RSSI value has a higher variation. The RSSI value can be calculated by the following simplified equation:

RSSI (dBm) = -0.6 * AgcLvl - 14,5

4.8 Energy Detection Level

After an energy detection scan, a value representing the maximum measured energy level is stored to the macScanED register. The stored value can be assigned to a receive signal power level. Figure 4.7 depicts the matching of the stored value in the register macScanED to the detected energy. Refer to *IEEE 8.15.4-2003 Std.* [1] section 6.7.7 for further information on energy detection.

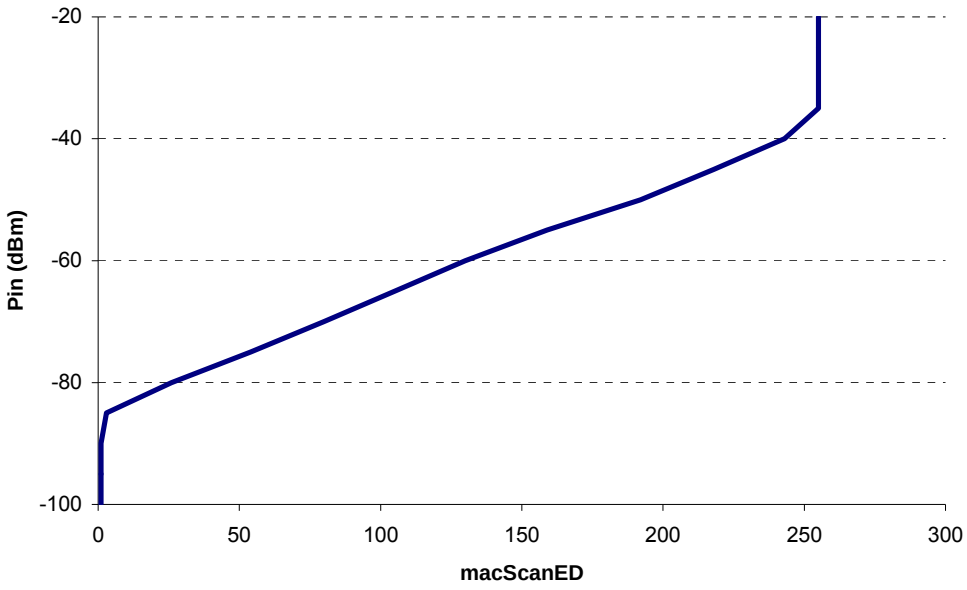


Figure 4.7: Detected Energy Level vs. macScanED register value

The detected energy can be approximated from the macScanED register value by the following equation

Pin / dBm = 0.185 · macScanED – 84.9 valid for 0 < macScanED < 255

Pin < -85 dBm valid for macScanED = 0

Pin > -30 dBm valid for macScanED = 255

4.9 Frame Filtering

The ZWIR4501 has an integrated receive frame filter with the following three filter levels:

- Level 1: CRC filter
- Level 2: Frame type and address field filtering according to section “7.5.6.2 Rejection and Reception” in the IEEE_standard
- Level 3: Operating-mode-dependent frame type filter

At the 1st level, frames with an incorrect CRC checksum are rejected.

In the 2nd level the frame type, address and PAN identifier are checked according to section “7.5.6.2 Rejection and Reception” in the IEEE_standard. The reference address and PAN identifier, against which the check is done, must be configured in the register bank section, starting from macPanId. By default, frames with reserved or not defined frame types are rejected by the level 2 filter. For forward compatibility however, frames with reserve frame types can be allowed by setting the ReservedFrameTypeEnable bit in the macFilterConfig register. In this case, the filter rules defined in the section “7.5.6.2 Rejection and Reception” in the IEEE_standard can also be applied to frames with a reserved frame type by enabling the ReservedFrameTypeFilterEnable bit in the macFilterConfig register.

The 3rd level is an operating-mode-dependent frame type filter. It performs the following filtering:

- Reject all non-acknowledge frames while waiting for an acknowledgment
- Reject all nonbeacon frames in beacon track mode during the beacon scan phase
- Reject all nonbeacon frames during active and passive scan
- Reject all non-command frames during orphan scan

By default all three filter levels are enabled. For debugging purposes or for promiscuous mode operation, the three different frame filter levels can be disabled independently in the macFilterConfig register. Frames that do not pass a filter level are rejected and automatically removed from the RxFIFO at the end of the reception.

For debugging purposes, an interrupt can be enabled in the IRQmask register indicating frames not passing the filter levels 1 and 2. Note that the frame filter interrupt is independent of the frame filter activation, e.g., the level 2 filter can be disabled in macFilterConfig, but an interrupt can still be generated for frames violating the level 2 check under the condition that the level 2 filter interrupt condition is unmasked in IRQmask.

4.10 Operation Mode Overview

The following section describes the different operating modes of the ZWIR4501 with the help of several SDL diagrams. Figure 4.8 shows the symbols used in these diagrams. The green input and output are the GPD and the IRQ, respectively. Most of the operating mode actions are initiated by the macControl command (violet input). Status information is held in the status register (violet output). The yellow symbols show the internal timer start/stop commands and expiration flags.

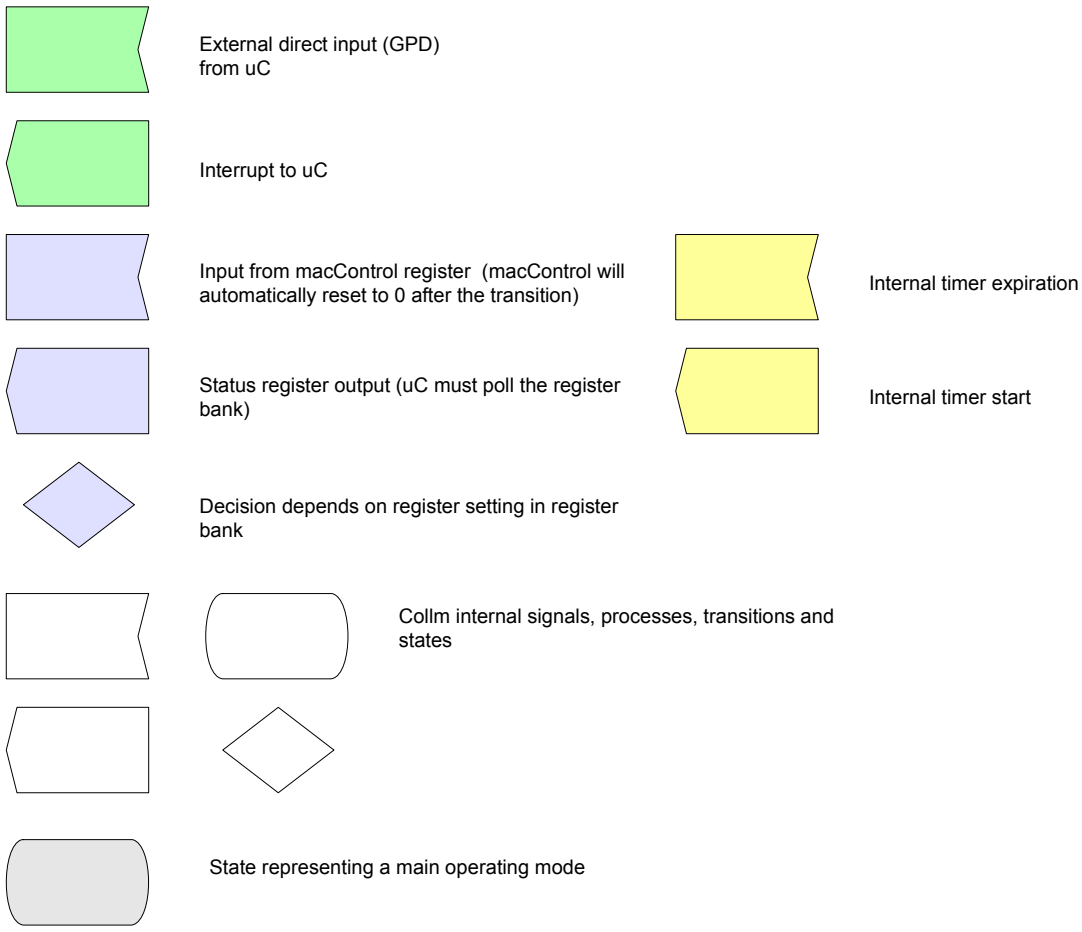

External direct input (GPD) from uC
Interrupt to uC
Input from macControl register (macControl will automatically reset to 0 after the transition)
Internal timer expiration
Status register output (uC must poll the register bank)
Internal timer start
Decision depends on register setting in register bank
Collm internal signals, processes, transitions and states
State representing a main operating mode

Figure 4.8: SDL Diagram Legend

This section begins with an overview of operating modes and the possible paths between them. The different operation modes are indicated as gray process symbols in the SDL diagram in Figure 4.9. Each operating mode is described in detail in subsequent sections. The following paragraph explains the SDL diagram in Figure 4.9 on page 35.

The ZWIR4501 powers up to Idle mode, from which all other operating modes can be entered. In Idle mode, the 24 MHz oscillator and the 32.768 kHz RTC are running and the analog front-end is shut down. Note that the 32.768 kHz RTC requires about 730 ms to be stable after the circuit power up (see section 3.5.1 for more information on power up). Therefore, functions relying on the Sleep mode and beacon tracking are available only after this start-up time. (See section 2.5.)
After power up it is recommended to read the trimming values including the transceiver's unique 64-bit IEEE address from the embedded EEPROM using the MTP procedure (see section 5.17 for further information on the recommended startup register setup).

A data transmission (Tx) is initiated with the macControl command TxOn. The frame MAC header and the MSDU must be stored in the appropriate Tx registers (see section 5.6) and the TxFIFO, respectively. The transmission ends after the successful transmission of the frame, or it is terminated by a channel access failure due to a busy channel or an unavailable/finished CAP or GTS period. After the end of the transmission, the ZWIR4501 either returns to Idle mode or automatically enters the receive mode (RxActive). The latter case occurs, if GoToRx is enabled in macTxConfig or a frame has been transmitted with the acknowledge request bit set in the frame control header (mhrFc1Tx).

From Idle mode, the receiver is turned on with the macControl command RxOn. The analog receiver is powered up and the acquisition phase is started (macRxStatus is RxActive). During reception, the received frame is stored in the RxFIFO. If the received frame has an invalid CRC or does not pass the integrated frame filter (see section 4.9), the frame is removed from the RxFIFO and the ZWIR4501 continues with a new acquisition cycle. If the frame is accepted, the ZWIR4501 can continue automatically in three different ways. If a frame with the acknowledge request bit set in the MAC header frame control field (mhrFcRx) has been received and the AutoAckEnable is asserted in macRxConfig, the Tx mode is started after a Rx/Tx turnaround and the acknowledge frame is transmitted with a space of T_Ack symbols after the received frame. The acknowledge frame is generated with the MAC header frame control byte [15:8] set to zero bits and a sequence number from the received frame. Software must set the frame control field [7:0] (mhrAckFc1Tx) before the start of the acknowledge transmission.

If the received frame does not request an acknowledgment or the AutoAckEnable in macRxConfig is not set, either the second or the third path is taken. If the ContRx bit in macRxConfig is not asserted, the receiver is powered down and the Idle mode starts. If the ContRx bit is set, the ZWIR4501 remains in RxActive mode and continues with a new acquisition cycle.

Additionally, the ZWIR4501 can be directed from RxActive mode into Tx, Idle, beacon tracking, or AutoBcTx mode with the appropriate macControl command. If reception is ongoing while one of these commands is received, the reception is first completed before the transition to the requested operating mode occurs. During this time, the command is buffered in the macControl register.

The four different scan modes are initiated with the macControl command ScanOn. The scan mode is selected in the macScanMode register. Some scan modes require additional macControl command interaction. For details, refer to section 4.11.4.
A scan process terminates either with a timeout or a macControl command ScanOff. After the scan process, the RxIdle state starts and the analog receiver remains on, but the digital receiver is stopped. From this state, a new scan cycle can be initiated by writing the macControl command TxRxOff, followed by the macControl command ScanOn. Before each scan, the channel must be set up in the RPCC. From RxIdle mode, the RxActive, Tx, BcTr and Idle modes can be accessed as well.

Beacon tracking (BeaconTrack) is initiated with the macControl command BcTrOn. This is possible only from Idle mode. After successful reception of the first beacon, the ZWIR4501 keeps track of the beacon interval and, for the following beacons, the BeaconTrack state starts automatically at a time T_BeaconScanStart (in symbols) before the expected arrival of the next beacon. The ZWIR4501 must be in one of the allowed operating modes indicated in Figure 4.9 on page 35 in order to enter beacon tracking automatically. Ongoing data receptions or transmissions will delay the transition to the beacon tracking mode.

To reduce the power consumption, the ZWIR4501 can also be directed to Sleep mode via the macControl command BcTrSleep. The ZWIR4501 will automatically wake up at the expected arrival of the next beacon.

The beacon tracking state is vacated after the successful tracking of a beacon or a SyncLoss timeout, and the RxIdle state starts. If a CSMA algorithm could not be completed in the previous CAP, the TxMode starts automatically after the next tracked beacon and the CSMA algorithm is resumed. If a “Wait for frame response” procedure could not be completed in the previous CAP, then the RxMode starts automatically after the next tracked beacon and the “Wait for frame response” procedure is resumed. Beacon tracking can be disabled with

the BcTrOff command. This command does not cause an operating mode change except during beacon tracking, in which case it causes the ZWIR4501 to enter the Idle mode.

Automatic beacon transmission is initiated with the macControl command AutoBcTxOn. A beacon frame is automatically transmitted at every T_BeaconInterval symbol. An interrupt is generated $2^{Td\_BeaconInterval}$ symbols before the end of the beacon interval, triggering to the software to prepare the next beacon MAC header and MSDU.

The ZWIR4501 must be in one of the allowed operating modes indicated in Figure 4.9 in order to enter AutoBcTx automatically. Ongoing data receptions are canceled before entering the AutoBcTx mode. After the transmission of the beacon frame, the ZWIR4501 automatically enters either the RxActive mode and starts receiving or the Tx mode in order to resume a CSMA process that could not be completed in the previous CAP. Automatic beacon transmission can be disabled with the macControl command AutoBcTxOff. This command does not cause an operating mode change except during AutoBcTx, where it causes the ZWIR4501 to enter the Idle mode.

The ZWIR4501 can be forced to Global Power Down mode at any time by setting GPD to high (see Figure 4.11). It remains in Global Power Down mode until GPD is released to low. The Sleep mode is entered with the macControl command Sleep. The sleep duration is defined by the T_General register. In nonbeacon-enabled operation, the T_General corresponds to the sleep duration followed by the macControl command Sleep. In beacon-enabled operation, T_General defines the wake up time relative to the end of the beacon interval. The ZWIR4501 can be forced to exit the Sleep mode by applying a high-to-low transition on the GPD input (see Figure 4.12).

For both Sleep and Global Power Down modes, the analog part and the 24 MHz crystal oscillator are shut down. Only the 32.768 kHz RTC continues to run. An exception arises if the clock output (CLKO) is configured to feed a signal derived from the 24 MHz clock during Global Power Down or Sleep mode (refer to ClkOutConfig register). Under this condition, the 24 MHz crystal oscillator is not shut down.

The ZWIR4501 provides two general purpose timer functions that can run in the background of the normal operating mode processes. There are three different timer sources available for this function (refer to section 4.13). The first function is the T_General timer, which can be programmed to generate a timer interrupt at the time configured in the T_General register. The second function can be used to release a MAC control command at a defined time T_macControl. The timer is initiated by writing a command to the macControlT register. Upon execution of the macControlT command, a timer interrupt is generated.

Operating Modes
Note: The states represent the operating modes of the ZMD44101 and are spaceholders for more complex underlying FSM's and processes).
At this level only the macOpMode changes are shown. The status register updates (macTxStatus, macRxStatus, macBcTrStatus, macAutoBcTxStatus, macScanStatus) and the Internal timer and transitions are not shown at this level. Refer to the detailed sub mode diagrams.

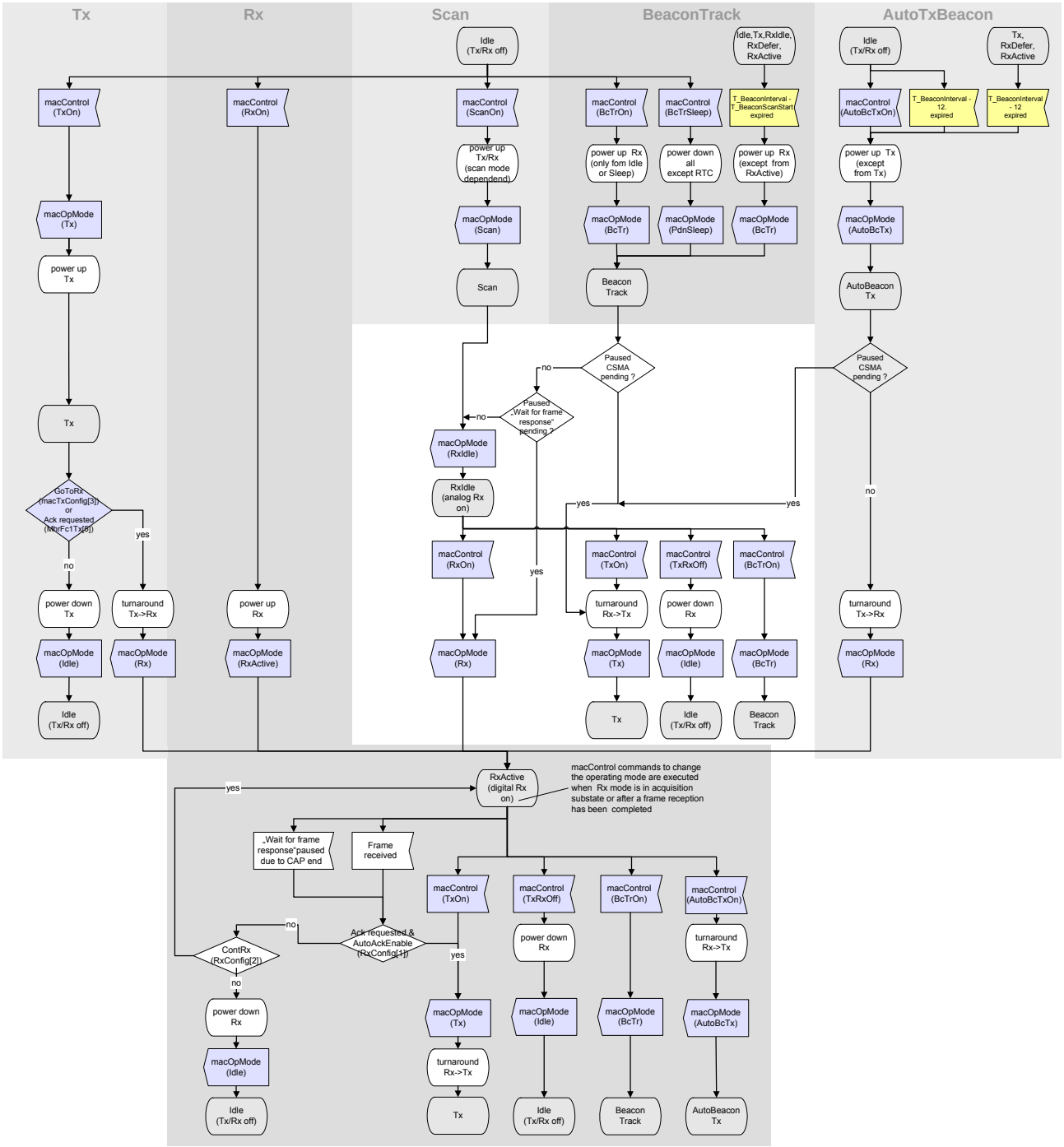


Figure 4.9: Operating Modes Overview SDL Diagram

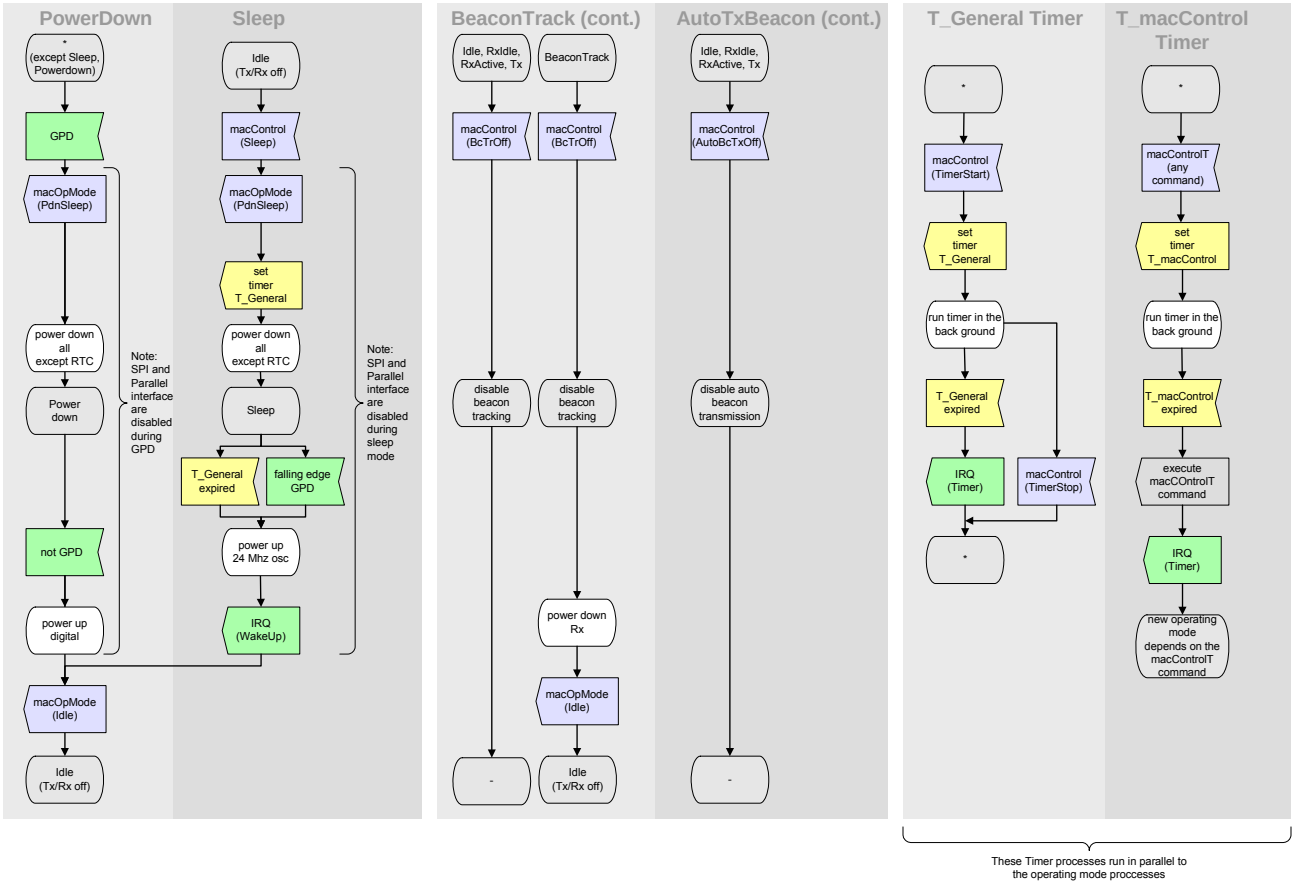


Figure 4.10: Operating Modes Overview SDL Diagram (cont.)

4.11 Operation Mode Description

The ZWIR4501 has five different modes of power management. These modes are user-configurable and controlled by the external microcontroller. The power modes are as follows:

- **Tx/Rx**: Tx or Rx mode is active. The analog front-end is powered up.
- **Idle mode:** The analog front-end is powered down, but the 24 MHz crystal oscillator remains on.
- **Sleep mode**: All circuits are switched off except the 32.768 kHz RTC for accurate time reference. Power consumption is reduced to 2.3 µA (typical).
- **Global Power Down mode:** The ZWIR4501 enters into Global Power Down mode by setting the Global Power Down (GPD) function. The 32.768 kHz RTC is maintained.
- **Off mode:** Both clocks and oscillators are shut down. The register content is not maintained. The Off mode must be reset to enter Idle mode correctly.

The ZWIR4501 has a Power On Reset (POR) function (refer to section 3.5.1 for further information on POR).

4.11.1 Low Power Modes

Global Power Down (GPD) Mode
The ZWIR4501 can be forced into Global Power Down mode at any time (requires runing RTC) by setting the GPD pin to high. It remains in Global Power Down mode until the GPD pin is released to low. Figure 4.11 shows the GPD sequence and its timing.

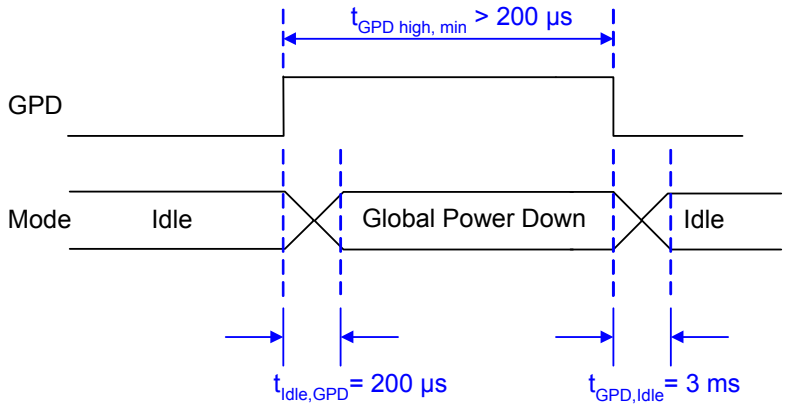


Figure 4.11: Global Power Down Sequence and Timing

In the Global Power Down mode, the analog part and 24 MHz crystal oscillator are shut down. Only the 32.768 kHz RTC continues to run. The SPI is fed from the 24 MHz clock, therefore the SPI is not accessible while the 24 MHz is shut down. The 24 MHz oscillator is not shut down, if the clock output (CLKO) is configured to feed a signal derived from the 24 MHz clock during Global Power Down mode (refer to section 3.4).

(Timer-controlled) Sleep Mode
The Sleep mode can be entered from Idle mode by the macControl Sleep command. The sleep duration is defined by the T_General register. In nonbeacon-enabled operation, the T_General corresponds to the sleep duration, followed by the Sleep command. The minimum sleep duration in nonbeacon-enabled operation is 3 ms. Because the sleep duration is configured in RTC units, the minimum value using Sleep mode for the T_General register is 0x32 in EU mode and 0x63 in US mode. Once Sleep mode has been entered, it is not possible to access the registers through the SPI or parallel interface. The ZWIR4501 can be forced to exit the Sleep mode before the sleep timer expires with a short GPD sequence. The short GPD sequence is shown in Figure 4.12.

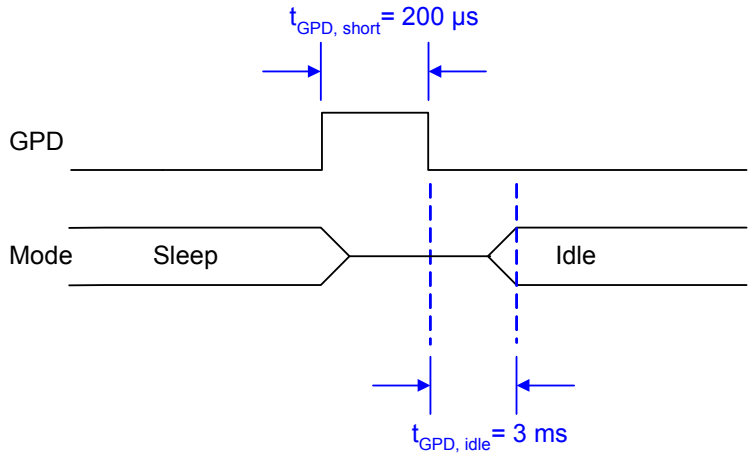


Figure 4.12: Short GPD Sequence to Immediately Leave the Timer-controlled Sleep Mode

In beacon-enabled operation T_General defines the wake up time relative to the end of the beacon interval. Slave devices in a beacon-enabled network can also sleep until the end of the beacon frame before they automatically return to beacon tracking mode. This beacon track sleep mode is activated with the macControl BcTrSleep command from the Idle mode.

In Sleep mode, the analog part and 24 MHz crystal oscillator are shut down. The SPI is fed from the 24 MHz clock, therefore the SPI is not accessible while the 24 MHz is shut down. Only the 32.768 kHz RTC continues to run. The 24 MHz oscillator is not shut down if the clock output (CLKO) is configured to feed a signal derived from the 24 MHz clock during Sleep mode (refer to section 3.4).

Off Mode

The lowest power consumption is achieved by using the transceiver's Off mode. During Off mode, both oscillators are switched off and therefore all functions based on the RTC (like beacon timing, timer-controlled sleep mode and GPD) are not supported until the RTC is stabilized again. The Off mode is enabled and disabled by the following procedures. It can only be entered if RTC is active.

Enter Off mode

- Write the value 0xBF to the RPD register to switch off the 32 kHz oscillator when setting the GPD to high.
- Set the GPD pin to high level to switch off both oscillators and thus also the digital core.

Leave Off mode (return to Idle mode)

- Set the GPD pin to low level to switch on both oscillators and thus also the digital core.
- Reset the ZWIR4501 with $t_{RSN\ hold\ 2}$ after which the digital core is active only again (RTC, timer functions and beacon mode are not available) and after 730 ms the RTC becomes active automatically, too. Or
- Reset the ZWIR4501 with $t_{RSN\ hold\ 3}$, after which the digital core and the RTC are active again.

Figure 4.13 shows the Off mode sequence and its timing.

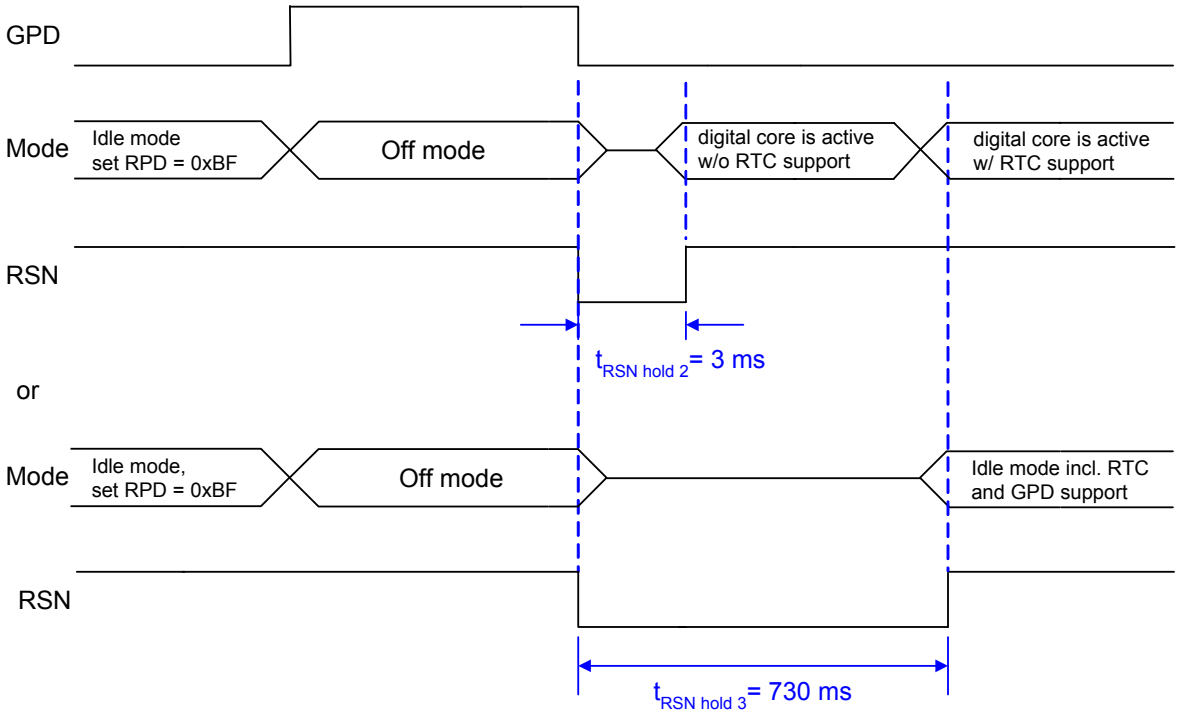


Figure 4.13: Off Mode Sequence and Timing

If the transceiver is used only for nonbeacon-enabled networks and the timer functionality, like Sleep mode, is not used (i.e., RTC is not used), the digital core and its interface (e.g. SPI) can be used after a reset with $t_{RSN\ hold\ 2}$ = 3 ms. The RTC core (including timer functionality) operates after 730 ms. The OFF mode can be entered again, if the RTC core is running.

4.11.2 Transmit Mode (Tx)

The ZWIR4501 supports five different transmit modes plus automatic acknowledge frame generation. The five transmit modes subdivide into the following modes.

Three slotted modes:

- GTS
- Slotted CSMA
- Slotted Direct Tx

Two unslotted modes:

- Unslotted CSMA
- Unslotted Direct Tx

The transmit mode is configured in the macTxConfig register.

A frame transmission is initiated with the macControl command TxOn. Before this command, the MHR, the MSDU and the MSDU length must be stored in the associated register (mhrFc1Tx, mhrFc2Tx and other address registers) and in the TxFIFO. Alternatively, the complete frame, including MHR and MSDU, can also be stored in the TxFIFO. For details about the frame handling, refer to section 4.5.

The ZWIR4501 maintains an internal inter-frame spacing timer (T_LIFS / T_SIFS) and the appropriate spacing before transmitting a frame.

In slotted CSMA and slotted direct Tx mode, the first symbol of the transmit frame is aligned with the boundary of a backoff period, where the first backoff period starts at the beginning of the superframe. The duration of a backoff period can be modified in the macUnitBackOffPeriod register but should be left at the default value for standard compliance. DirectTx mode can be used for direct transmission without going through the CSMA algorithm.

In the case of a successfully completed transmission, the Tx mode returns the macTxStatus as success. The status CAfail_CHbusy indicates that transmission failed because the CCA could not sense the channel as idle. The CSMA algorithm is described in section 4.11.2.1.

Each slotted CSMA transmission starts with a random backoff time. If the random backoff is not finished within the current contention access period (CAP) or within the 1st six slots following the beacon IFS for the battery life extension mode, then it is paused, and the macTxStatus CAfail_CAPfail is returned. The unfinished random backoff is automatically resumed in the next CAP. After the random backoff, the HW-MAC checks whether the current transaction, including a possible acknowledgment, can be completed before the end of the CAP. If the check fails, the frame is not transmitted and the returned macTxStatus is CAfail_CAPfail. Under this condition the CSMA will automatically resume with a new random backoff in the next CAP.

For guaranteed time slot (GTS) transmission, the transmitter checks whether the guaranteed time slot is still available in the current superframe and if so, whether the transmission, including acknowledgement, can be completed before the end of the GTS. If one of these checks fails, the frame is not transmitted and the returned macTxStatus is GTSfail.

Figure 4.14 on page 40 illustrates the GTS and CAP boundary checks. A CAP/GTS transmission is done only if the complete transaction, including acknowledgment, can be completed within a long/short inter-frame space (T_LIFS / T_SIFS), plus a guard time (T_Delta), before the end of the CAP or GTS. The guard time is used to

compensate for the timing deviation between a device and its coordinator introduced by the different crystal frequency offsets. It is significant only for higher superframe orders. By default, this guard time is disabled. If required, it can be enabled with the LateGuard bit in the macBcTrConfig register.

A GTS transmission can also be delayed by the T_Delta time in order to compensate for a device's early timing offset relative to its coordinator. Again, this effect is significant only for higher superframe orders and the early guard time must be enabled in the macBcTrConfig. For more details on network timing issues, refer to section 4.12, 4.12.2 and 4.12.3.

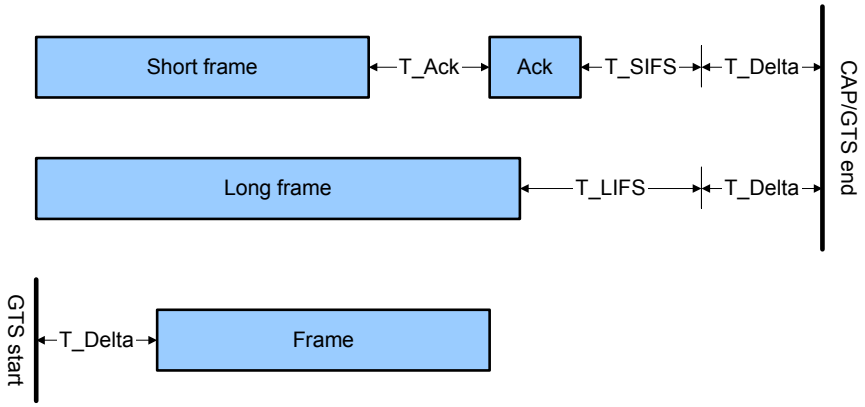


Figure 4.14: GTS and CAP Check

At the end of a transmission, there are several ways that the Tx mode can be exited. First, if beacon tracking or automatic beacon generation is activated and the T_BeaconInterval timer has expired, the ZWIR4501 will automatically enter the automatic beacon transmission or beacon tracking mode. This timer expiration has the highest priority.
Second, after the successful transmission of a frame with the acknowledge requested bit set, the ZWIR4501 automatically activates its receiver and waits for a period T_WaitForAck in symbols for the reception of an acknowledge frame.
Third, if acknowledgment is not requested, the appropriate inter-frame spacing timer is started at the end of the transmit frame and, depending upon the GoToRx bit in the macTxConfig register, either the Rx mode starts or the ZWIR4501 returns to Idle mode. Fourth, in all remaining cases, either the Rx mode or the Idle mode starts, depending upon the GoToRx bit.

The Tx mode is also used to generate acknowledge frames automatically, if they are requested by a received frame and the AutoAckEnable bit is set in the macRxConfig register (entry point (C) in Figure 4.15). The ZWIR4501 maintains the spacing T_Ack between the frame and the acknowledge frame. At the end of the acknowledge transmission and depending upon the GoToRx bit, either the beacon tracking or automatic beacon transmission mode starts (if the T_BeaconInterval timer has expired) or the Rx mode or Idle mode starts depending on the GoToRx bit. The inter-frame spacing timer starts at the end of the acknowledge frame.

If the macControl command TxRxOff is sent to the transceiver while a frame is transmitted, an interrupt with the IRQreason 0x01 (with an associated macControl value of 0x1F) is issued indicating that a command error has occurred. The frame transmission is completed and the transceiver is switched to receive mode if the automatic acknowledgment feature is enabled with the AutoAckEnable bit in the macRxConfig register. If the macControl command TxRxOff is sent after a frame and before the expected acknowledgment frame, the reception of the acknowledgment is cancelled. In such a situation, no receive interrupt is issued and the macRxStatus register reports that the receiver is switched off by containing the value 0x00. The macControl command TxRxOff does not have any influence on the acknowledgement frame reception once the reception of the acknowledgment has been started. In this case, the entire acknowledgement frame is received, and an interrupt is generated with IRQreason 0x02, macRxStatus 0x10.

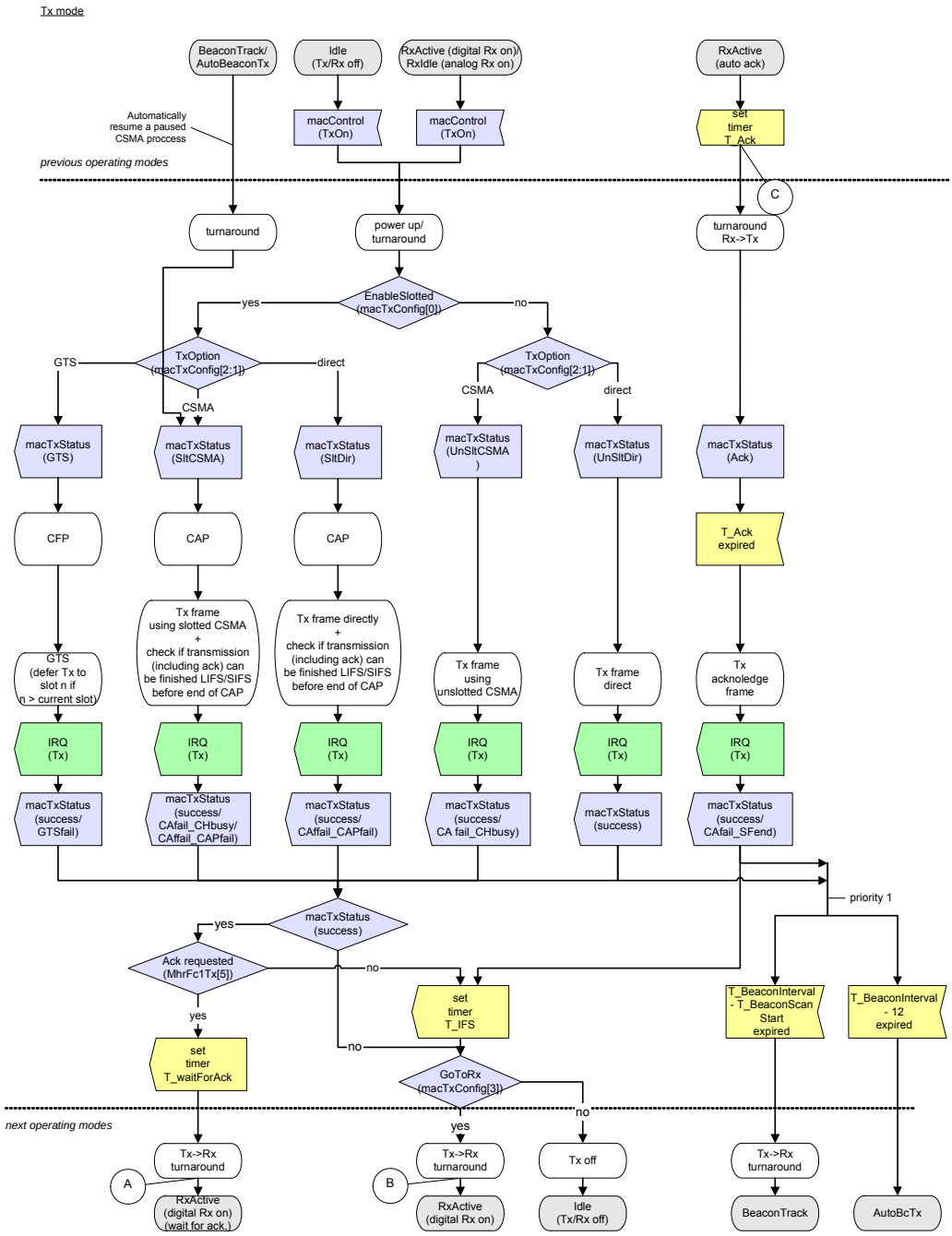


Figure 4.15: Tx Mode SDL Diagram

4.11.2.1 CSMA-CA Algorithm

The CSMA-CA algorithm is defined in the standard *IEEE 802.15.4, Section 7.5*. The algorithm is configured by a set of registers (macMinBE and following) indicated in italics in Figure 4.16 on page 42. These values should be changed with care, since they affect standard compliance.
In slotted mode, the CSMA algorithm is automatically paused if it cannot be finished in the current CAP or in the first six slots following the beacons IFS. in battery life extension mode. If the CSMA algorithm is paused, then a Tx IRQ with the macTxStatus CAfail_CAPfail is returned. A paused CSMA process will be resumed automatically in the next CAP.

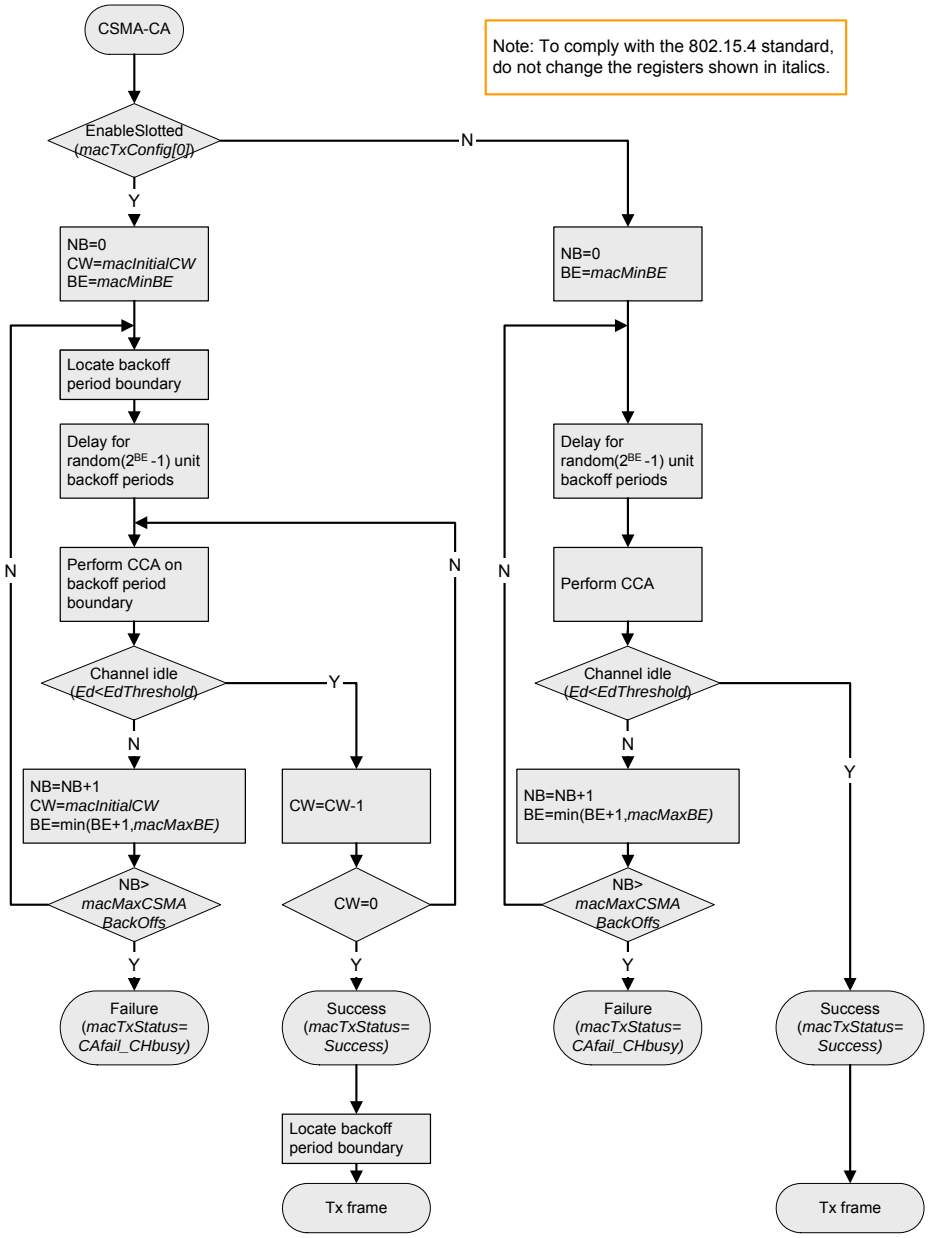


Figure 4.16: CSMA-CA Algorithm

4.11.2.2 MHR Address Field Composition

The transmit frames are formed within the framer module of the ZWIR4501. The MHR is taken from the appropriate registers (mhrFc1Tx and following); the MSDU is taken from the TxFIFO and appended to the MHR, and the frame is then appended to the CRC field and fed into the PHY layer. The address field content and length within the MHR depends upon the different addressing mode and frame type settings of the frame control field (mhrFc1Tx and mhrFc2Tx). The different address field combinations are shown in Figure 4.17.

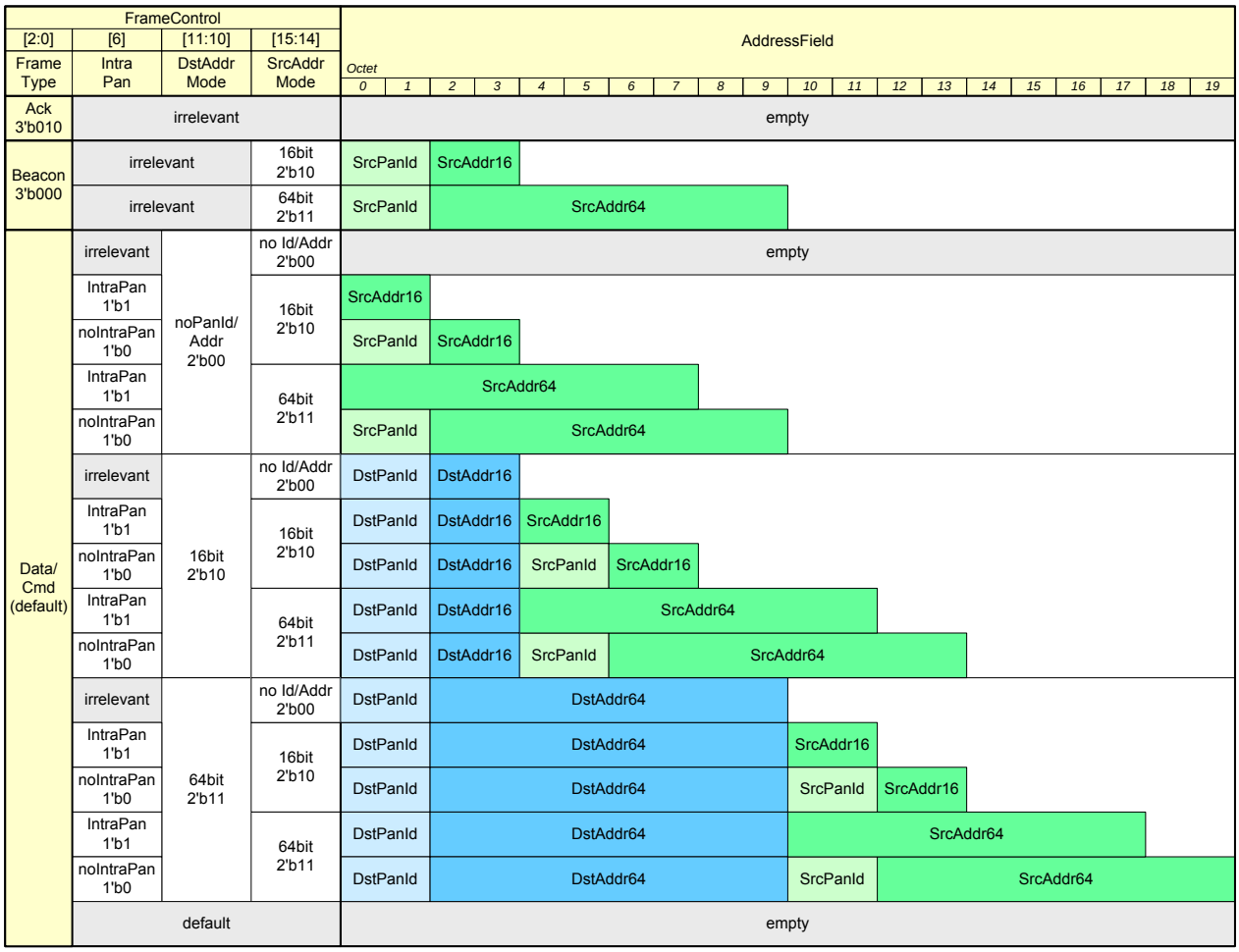


Figure 4.17: MAC Header Address Field Composition

4.11.3 Receive Mode (Rx)

The Rx mode can be entered by either the macControl command RxOn, an autonomous turnaround coming from Tx or AutoBcTx mode (entry B, Figure 4.18) or from Tx mode in order to receive an acknowledgment (entry A, Figure 4.18), or automatically from BeaconTrack or AutoBeaconTx mode in order to resume a paused “Wait For Frame Response” cycle (entry D, Figure 4.18).

The macControl command RxOn can be issued in Idle or RxIdle mode. After the receiver has powered up, the acquisition phase starts. The macRxStatus is Acquire. If a frame has been detected, it is received and stored into the RxFIFO (macRxStatus = Rx). As explained in section 4.5, a copy of the MHR and CRC is also stored in the register bank. At the end of the reception, the CRC sequence and the frame filter are checked.

If the CRC checksum is invalid or does not pass the frame filter rules, the frame is removed from the RxFIFO and the acquisition is resumed. Reception of a correct frame is indicated by an interrupt and the appropriate IRQreason. More detailed information can be derived from the macRxStatus register. For correct frames, the status is Data or DataAck if the received frame requests acknowledgment.

After a correct frame, there are three ways to proceed. If acknowledgment is requested and AutoAckEnable is asserted in macRxConfig, the Tx mode starts (exit C, Figure 4.18) and an acknowledgment containing the sequence number of the received frame is transmitted. The HW-MAC develops the frame to acknowledge spacing T_Ack. Software must set the frame pending bit in the frame control field (mhrAckFc1Tx) within this period. If the received frame does not request an acknowledgment or the AutoAckEnable bit is not set, either the analog receiver is powered down and the ZWIR4501 returns to Idle mode or, depending on the ContRx bit in the macRxConfig register, a new acquisition cycle starts. The interframe spacing timer (T_LIFS / T_SIFS) is started at the end of the received frame.

If the receiver starts from the Tx mode (entry A, Figure 4.18), the HW-MAC waits T_WaitForAck symbols for the reception of an acknowledge frame. If no acknowledge frame is received within this time, an Rx interrupt is generated with the associated macRxStatus AckTimeOut. If an acknowledgment has been received in time, the acknowledge frame is stored in the RxFIFO. The MHR is also stored in the register bank. Storage of the acknowledge frame in the RxFIFO can be disabled in the macRxConfig register. The ZWIR4501 provides support for the MLME-POLL primitive, which can be enabled by the WaitResponseEnable bit in macRxConfig. If this support is enabled and the received acknowledgment has the frame pending bit in the MHR frame control field set, the HW-MAC will start a timer (T_MaxFrameResponse) and wait for the reception of a frame. If no data is received within this time, an Rx interrupt is generated and the macRxStatus is PollNoData. In slotted mode the T_MaxFrameResponse timer will count only CAP symbols. If the polled data is not received within the CAP and the T_MaxFrameResponse has not expired, an Rx interrupt is generated and the macRxStatus is PollCAPend. In this case, the T_MaxFrameResponse is paused and the polling process is automatically resumed in the next CAP (entry D, Figure 4.18).

There are several additional ways to exit the Rx mode via macControl commands or timer expirations. It is important that an ongoing reception is not interrupted before exiting the Rx mode. The expiration of the T_BeaconInterval timer has higher priority than any other commands, and it causes the ZWIR4501 to enter either beacon tracking or automatic beacon transmission mode automatically, depending upon which mode is active. The switch to automatic beacon transmission cancels ongoing receptions. The macControl commands TxRxOff, AutoBcTxOn and TxOn also cause the ZWIR4501 to exit the Rx mode and to enter Idle, AutoBcTx or Tx mode, respectively.

Note that the macRxStatus registers are separated into two concurrent sections. One section indicates the current processing state, and the other shows the result of the last reception. For instance, if a frame has been received and the receiver continues in reception mode, the status will be 0x81, which is an OR combination of the processing state Acquire and the result indicator Data. A description of the receive frame handling and RxFIFO management is given in section 4.5.

If the macControl command TxRxOff or TxOn is issued while receiving a frame, the remaining bytes of the frame are received first, then the macControl command is executed. Following this sequence, a receive interrupt IRQreason 0x02 is generated; the corresponding macRxStatus reports three possible results: 0x00, 0x01, or 0xC0. In order to determine if the frame was received successfully, the macFramePend register must be checked. If the macFramePend register indicates a frame was received, the frame can be read from the RxFIFO or discarded using the macControl command RxFifoFlush.
If the inbound frame's field control field had the Ack requested bit set and AutoAckEnable bit in the macRxConfig register is set, the acknowledgment frame is transmitted automatically if the macRxStatus equals 0xC0. The acknowledgment frame is not transmitted if the macRxStatus equals 0x00 or 0x01.

See section 4.11.2 on page 40 for further information on acknowledgment frame reception while issuing the macControl command TxRxOff.

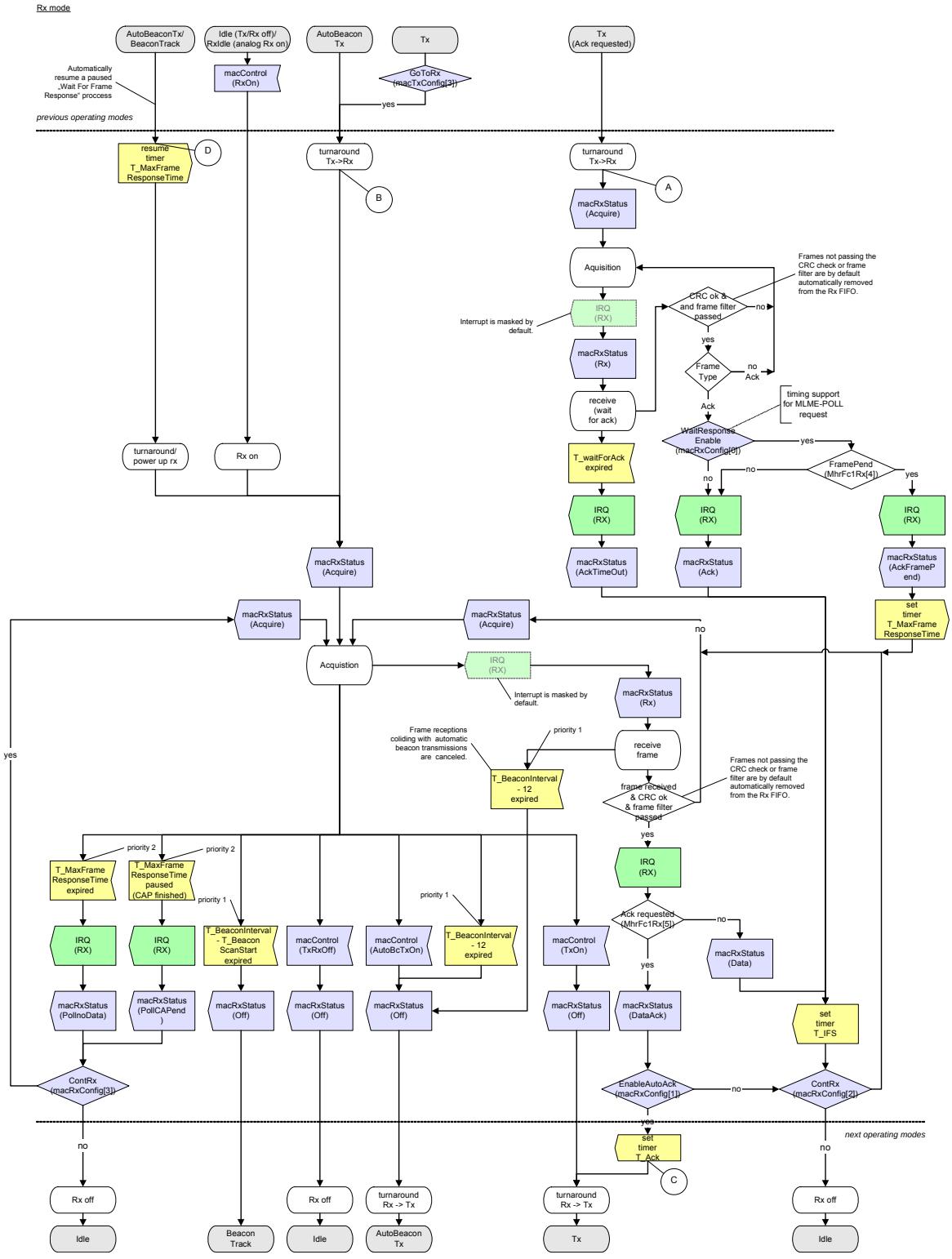


Figure 4.18: Rx Mode SDL Diagram

4.11.4 Scan Modes

The ZWIR4501 supports the four scan modes: energy detection (ED), active, passive and orphan scan. The scan mode is selected in the macScanMode register. A scan can be activated only from the Idle mode with a macControl command ScanOn. Before the macControl command ScanOn is issued, the following registers need to be initialized: write value 0x10 to register address 0x43 and write value 0x00 to register addresses 0x44, 0x45 and 0x46. It is important, that register address 0x46 is written last.
For all four scan modes, the scan duration in symbols is set up in the T_ScanDuration register.

During an ED scan, an 8-symbol-long energy detection is performed repeatedly until the scan duration expires. The active ED scan is indicated by macScanStatus ED. At completion of the scan, an interrupt with the IRQreason Scan and the macScanStatus TimeOut is generated. The maximum recorded ED value during the ED scan can be read from the macScanED register. For details on the ED level refer to section 4.8.

The active scan mode allows a device to locate coordinators transmitting beacon frames. In active scan, a beacon request frame is transmitted first. First, the MHR and MSDU must be stored in the register bank and the TxFIFO, respectively. The macControl command ScanOn initiates the transmission of the frame using CSMA-CA. During this phase, macScanStatus is ActiveTx. If the CSMA-CA fails due to a busy channel, a scan interrupt is generated, and the macScanStatus is ActiveTxFail. If the transmission is successful, the HW-MAC activates the receiver and waits for T_ScanDuration symbols for the reception of a beacon frame. During this phase, the macScanStatus is ActiveRx. Nonbeacon frames are ignored during the scan. Upon reception of a beacon frame, an interrupt is generated and the macScanStatus is Beacon. At this point, the scan process can either be terminated via the ScanOff command or be continued by the ScanCont command until a specific number of beacons have been found. The T_ScanDuration timer expiration always terminates the scan process, which is indicated by an interrupt and a macScanStatus TimeOut indication.

The passive scan mode is similar to active scan except that transmission of the beacon request frame is skipped and the receive/scan phase is entered directly. The passive scan process is indicated by the macScanStatus being Passive.

The orphan scan allows a device to attempt to relocate its coordinator following a loss of synchronization. It is similar to the active scan mode except that, to start, an orphan notification command must be transmitted and then the HW-MAC waits for the reception of a command frame. If a command frame is received, software must check it and then instruct the HW-MAC to continue or stop the scan process. The macScanStatus indications appropriate to the orphan scan mode are OrphanTx, OrphanRx, OrphanTxFail and Command.
If the received command frame requests an acknowledgment and the AutoAckEnable bit is set in the macRxConfig register, then the acknowledge frame will be sent automatically by the HW-MAC. After transmission of the acknowledge frame, a Tx interrupt is generated and the macTxStatus is Success.

Once a scan cycle has been terminated, either by the macControl command ScanCont or a scan duration expiration, the RxIdle state starts. In RxIdle, the analog receiver is still powered up, but the digital circuit is idle. From this state, the scan mode can be resumed. The macControl commands TxOn, RxOn and TxRxOff cause the ZWIR4501 to switch into Tx, RxActive or Idle mode, respectively. In order to start a new scan cycle at another RF-channel, the HW-MAC must first be set to Idle mode via the macControl command TxRxOff. Then the new channel must be set up by the RPCC and RxMode registers. After a waiting time of approximately 200 µs, required by the PLL to settle to the new channel, the new scan cycle can be initiated via the macControl command ScanOn.

Note that the active, passive and orphan scan processes use the HW-MAC Rx and Tx functions. Therefore, in active and orphan scan, the transmission result is also indicated by a Tx interrupt, together with the appropriate macTxStatus equal to Success or CAfail_CHbusy. During the ScanRx phase, a successful data reception and a CRC failure are also indicated by the Rx and CRCfail interrupts. If these additional interrupts are not required, the user can mask them in the IRQmask register.

A detailed description of the scan modes from standard context can be found in *IEEE 802.15.4, Section 7.5.2.1*.

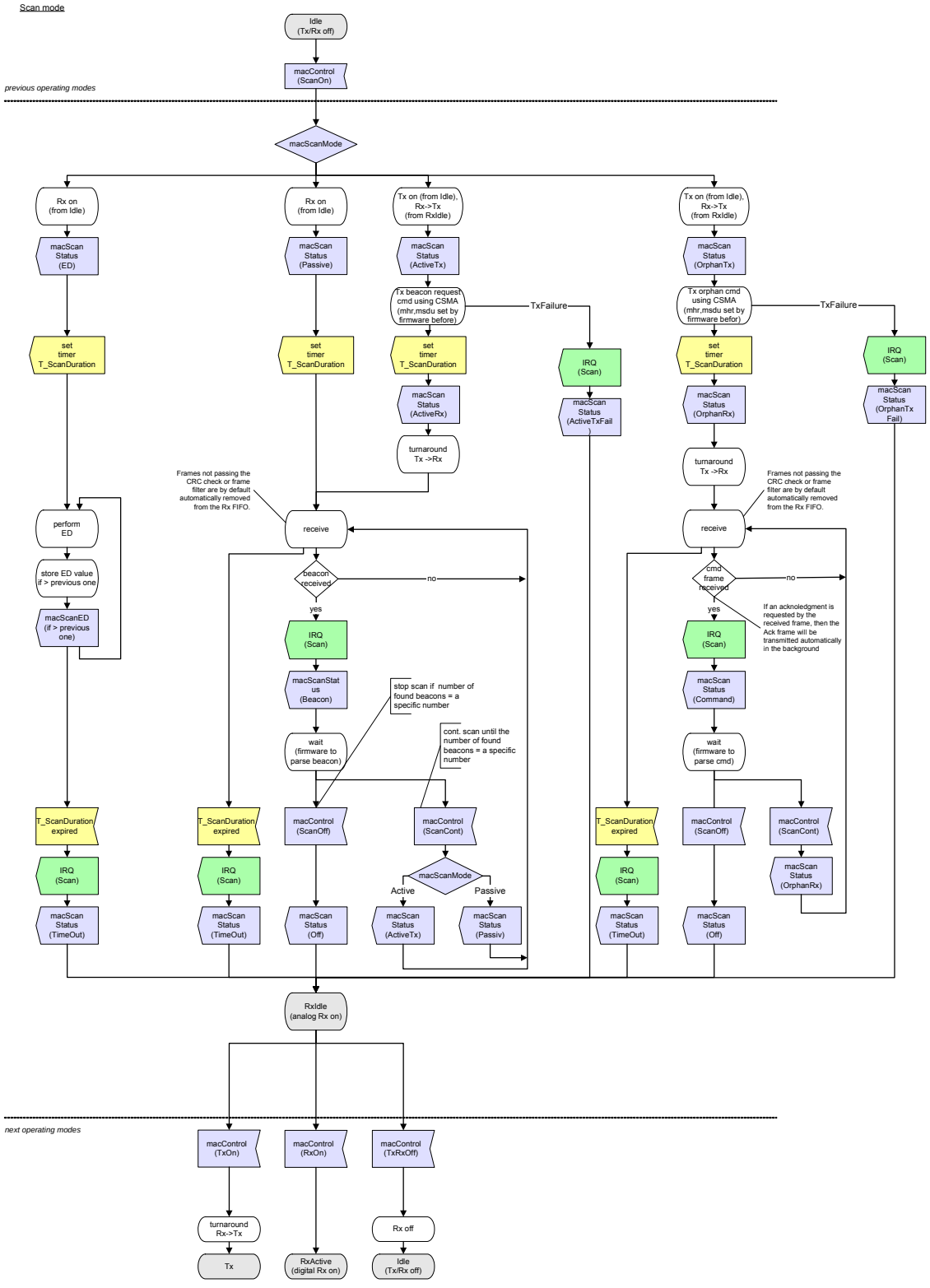


Figure 4.19: Scan Mode SDL Diagram

4.11.5 Automatic Beacon Generation

The automatic beacon generation (AutoBcTx) mode is used for coordinators to transmit beacon frames periodically. The automatic beacon generation must be initiated once via the macControl command AutoBcTxOn, after which the ZWIR4501 transmits a beacon frame every T_BeaconInterval symbols. At a time $2^{Td\_BeaconInterval}$ before the end of each beacon interval, an interrupt is generated with the IRQreason AutoBcTx requesting the software to prepare the MHR and the MSDU of the next beacon frame. The macAutoBcTxStatus sequence is Tx while transmitting the beacon, then Done after the beacon transmission and finally TxReady between the interrupt and the transmission of the consecutive beacon. If required, an additional interrupt indicating the completed beacon transmission can be enabled in the IRQmask register.

The active portion of the superframe within the beacon interval is configured by the macSuperframeOrder (SO; refer to section 4.11.7 for further information regarding superframe configuration). The current time in symbols within the superframe and the ongoing network time are monitored by the macCurrentSymbolTime and macTotalTimeFFD registers. The macCurrentSymbolTime is zero at the beginning of the superframe and counts up until the end of the superframe and then reverts back to zero until the next superframe. The beacon interval is derived from the 24-bit-wide T_TotalTimeFFD timer. This timer can be started either by the macControl command TotalTimerStart or, if it is not running, automatically when the automatic beacon generation is started. Each transmitted beacon is time-stamped at the leading edge. The timestamp is taken by default from the T_TotalTimeFFD register and can be read from the macBeaconTxTime register.
The automatic beacon generation and the related timing and status register are shown in Figure 4.20 below.

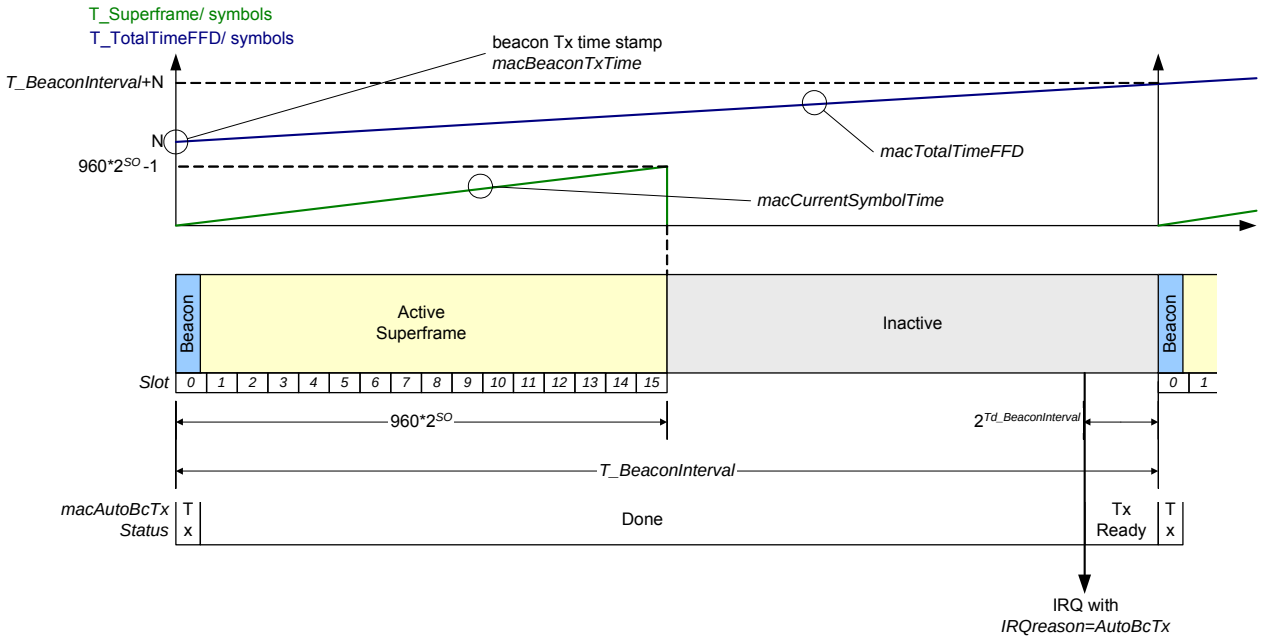


Figure 4.20: Automatic Beacon Generation

The automatic beacon generation can be initiated only from the Idle or RxActive mode. Once activated, the HW-MAC automatically returns to the automatic beacon transmission mode to transmit a beacon frame at the end of each beacon interval. The AutoBcTx mode starts $2^{Td\_BeaconInterval}$ symbols before the end of the beacon interval. This transition has priority over all other operating modes. For data transmission, the HW-MAC checks before each transmission begins to confirm that a transaction can be finished an inter-frame spacing time before the end of a CAP/GTS, so that these transactions cannot interfere with the beacon interval end. Ongoing data receptions are canceled before the ZWIR4501 enters the automatic beacon transmission mode.

After the beacon transmission, the HW-MAC automatically enters the Rx mode and starts the receiver, unless there is a paused CSMA process pending from the previous CAP. In this case, the TxMode starts automatically and the CSMA procedure is resumed.

The automatic beacon transmission can be disabled via the AutoBcTxOff. This command can be used in Idle, RxIdle, RxActive, and Tx modes. The operating mode remains unaffected in this case.

A TxFIFO usage conflict can occur if the TxFIFO is still occupied with data from a pending/ongoing transmission when software needs to write the MSDU of the next beacon into the TxFIFO. To reslove this problem, the RxFIFO can be used alternatively for the beacon transmission. This mode is enabled by setting the UseRxFIFO bit in the macBcTxConfig register. In this case, the integrated HW-MAC frame composition is bypassed and the complete beacon frame content, including MHR and MSDU, must be written to the RxFIFO (see section 4.9).

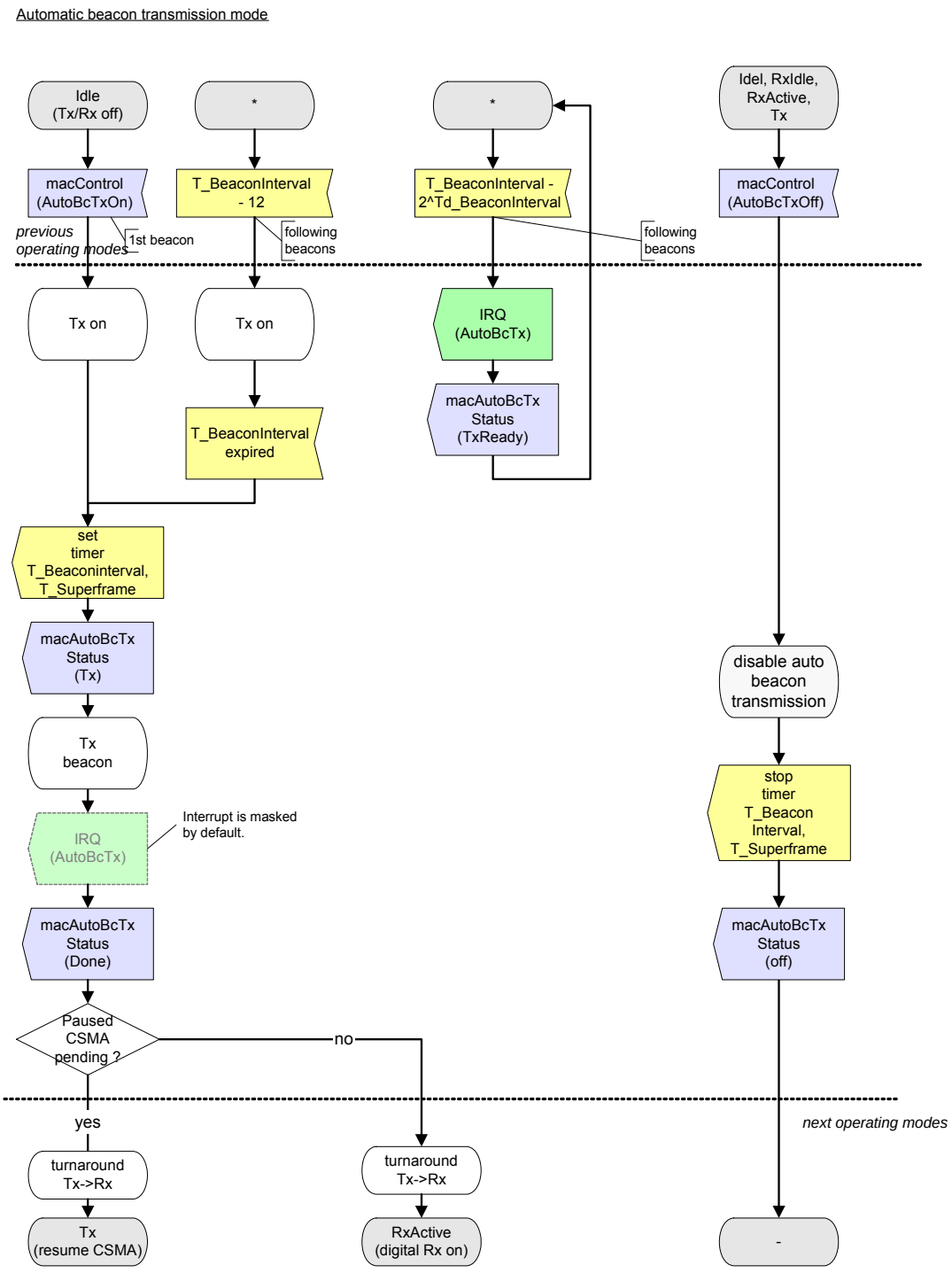


Figure 4.21: Automatic Beacon Generation Mode SDL Diagram

4.11.6 Beacon Tracking

The beacon tracking mode is used by devices in a beacon-enabled network to track the beacon frames of the network coordinator. The ZWIR4501 maintains the network time by its 32.768 kHz RTC, allowing the ZWIR4501 to go to Sleep mode within a superframe and wake up just before the next beacon arrival. The crystals' frequency offsets can cause a timing deviation between the coordinator and the device. On the device side, the ZWIR4501 keeps track of this deviation and corrects the beacon interval accordingly.

As shown in Figure 4.22, there are several timing registers involved in beacon tracking mode. The beacon interval in symbols is set up in T_BeaconInterval. For the beacon scan phase, two parameters must be adjusted. The first parameter is T_BeaconScanDuration, which defines the time period during which the receiver is turned on and the HW-MAC waits for the reception of a beacon frame before it is declared lost. The second parameter is T_BeaconScanStart, which defines the time in symbols during which the receiver is turned on and the new scan process is started before the arrival of the next beacon. It is not used for the initial beacon acquisition scan. For the following beacon frame, this parameter must be greater than the timing deviation introduced by the crystal frequency deviation. From this point on, the ZWIR4501 continues tracking the timing error until a beacon frame is missed. Thereafter for subsequent beacon scans, this start time can be reduced. Since there will be a residual timing error, T_BeaconScanStart must not be set to less than 5. The timing error effect becomes significant only for higher beacon orders. For details on the network timing, refer to sections 4.12 and 4.12.

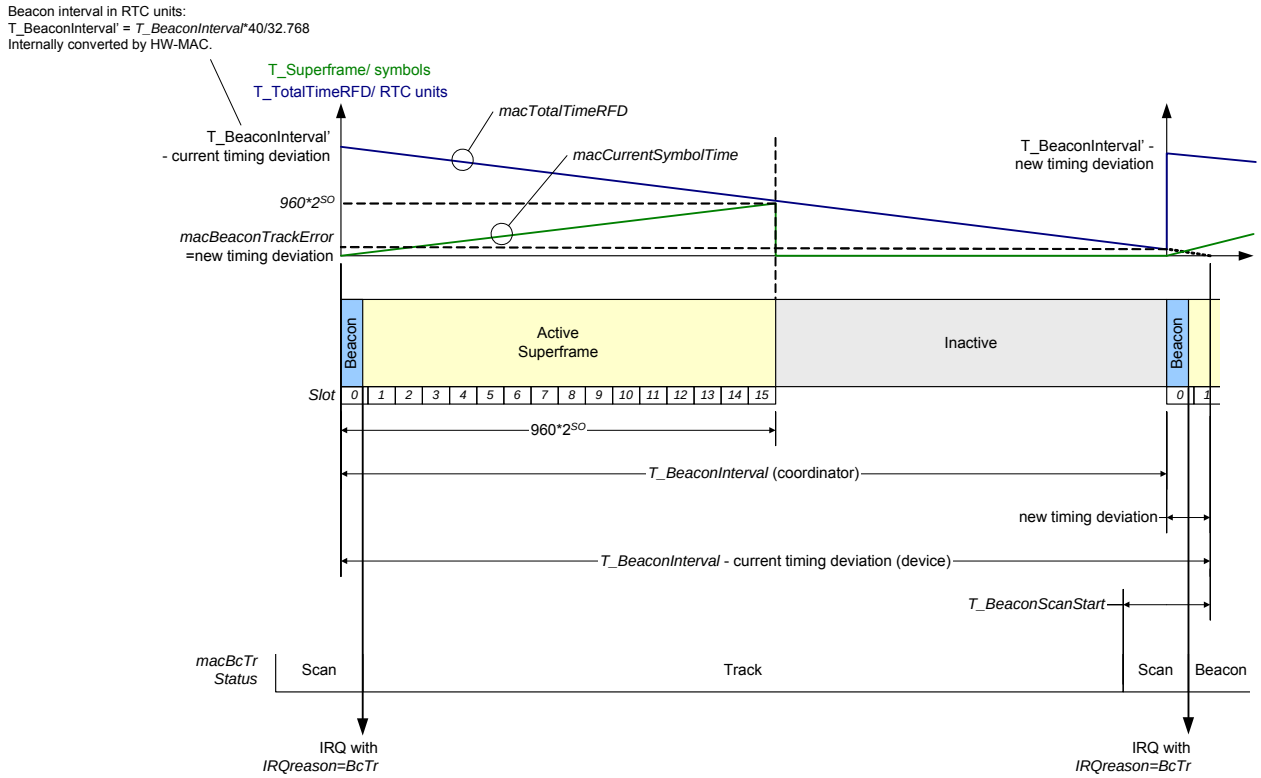


Figure 4.22: Beacon-Tracking Timing

The size of the active superframe is adjusted by the macSuperframeOrder (SO). The network time is monitored by the macTotalTimeRFD register in RTC units (1/(32.768 kHz) US, 2/(32.768 kHz) EU). The start value is the beacon interval minus the estimated timing deviation. The value zero is assigned to the expected arrival time of the next beacon frame. The arrival time of each tracked beacon frame can be read from the macBeaconRxTime register. It is a signed value in RTC units indicating the deviation from the expected arrival time. A positive value signifies early arrival and a negative value signifies late arrival. The symbol time within the active superframe is

indicated by the macCurrentSymbolTime in symbol units, where zero is the beginning of the superframe. The current slot number can be read from macCurrentSlot.

Figure 4.23 on page 54 shows the SDL diagram of the beacon tracking mode. Beacon tracking can be activated only from Idle mode via the macControl command BcTrOn. It starts with an initial scan for a beacon frame. All nonbeacon frames are ignored during this phase. If TrackEnable is not set in the macBcTrConfig and no beacon frame is received within T_BeaconScanDuration, or if TrackEnable is set and no beacon is received in a number macMaxLostBeacons of consecutive beacon scan phases, then a loss of synchronization has occurred. The synchronization loss with the coordinator is indicated by a beacon track (BcTr) interrupt and a macBcTrStatus of SyncLoss. After the synchronization loss, the RxIdle state starts.

The arrival of a beacon frame within the scan duration is indicated by a BcTr interrupt and a macBcTrStatus of Beacon. The PAN identifier and source address of the beacon are verified, together with the checksum, by the integrated frame filter at the end of the beacon reception. Beacon frames that do not pass the filter are rejected and automatically removed from the RxFIFO. If required, the firmware can do an additional check of the beacon frame. Therefore the BeaconConfirm bit must be set in the macBcTrConfig register. In this case, firmware can verify the beacon and either accept or reject it via the macControl commands BcOk and BcFail.

The HW-MAC synchronizes to beacons that are not rejected. After having synchronized to a beacon, the HW-Mac will keep track of the beacon. The beacon tracking mode is enabled by the TrackEnable bit in the macBcTrConfig register. The active tracking is indicated by the macBcTrStatus Track.
After the reception of the beacon frame, three different operating modes can be entered. If an unfinished CSMA procedure is pending from the previous CAP, the Tx mode will be entered automatically in order to resume the CSMA process. If an unfinished “Wait for frame response” process is pending from the previous CAP, then the Rx mode will be entered in order to continue with the “Wait for frame response” process. Otherwise, the RxIdle mode starts. The RxIdle is an interim state and firmware must direct the HW-MAC into an appropriate operating mode.

If beacon tracking is active, the ZWIR4501 automatically returns to the beacon tracking mode T_BeaconScanStart symbols before the estimated end of the beacon interval and starts to scan for the next beacon for a time T_BeaconScanDuration. This automatic switch can be performed from the operating modes Idle, RxActive, RxIdle, RxDefer and Tx. Ongoing transmission or receptions are completed first.
The macBcTrStatus register contains two additional bits indicating the current status of the integrated beacon interval timing correction. After each initial beacon, i.e., the first beacon after starting the beacon tracking or the first beacon received after a synchronization loss, the Sync bit will be set in the macBcTrStatus register. For each subsequent successfully tracked beacon, the Align bit and the Sync bit will be set in the macBcTrStatus register, indicating that the HW-MAC is keeping track of the coordinators beacon interval and that the internal beacon interval timing correction is active. For details about the integrated timing correction, refer to section 4.12 and 4.12.

From Idle mode, the ZWIR4501 can be sent into Sleep mode by the BcTrSleep command to reduce power consumption. The analog part including the 24 MHz crystal oscillator is powered down. The ZWIR4501’s 32.768 kHz RTC keeps track of the network time and wakes up and returns to the beacon tracking mode T_BeaconScanStart symbols before the estimated end of the beacon interval in order to track the next beacon frame. The wake-up is indicated by a WakeUp interrupt.

As mentioned above, the RxIdle state starts after the scan phase. In RxIdle, the digital core is idle but the analog receiver is powered up. This enables a fast receiver start via the macControl command RxOn. The Tx and the Idle modes can be initiated via the macControl command TxOn and TxRxOff commands, respectively. For power efficiency, use of the macControl command TxRxOff is recommended after a tracked beacon.

Beacon tracking can be deactivated with the macControl command BcTrOff from Idle, RxIdle, RxActive, Tx and BeaconTrack mode. From BeaconTrack, the Idle mode starts. For the other modes, BcTrOff does not produce an operating mode change.

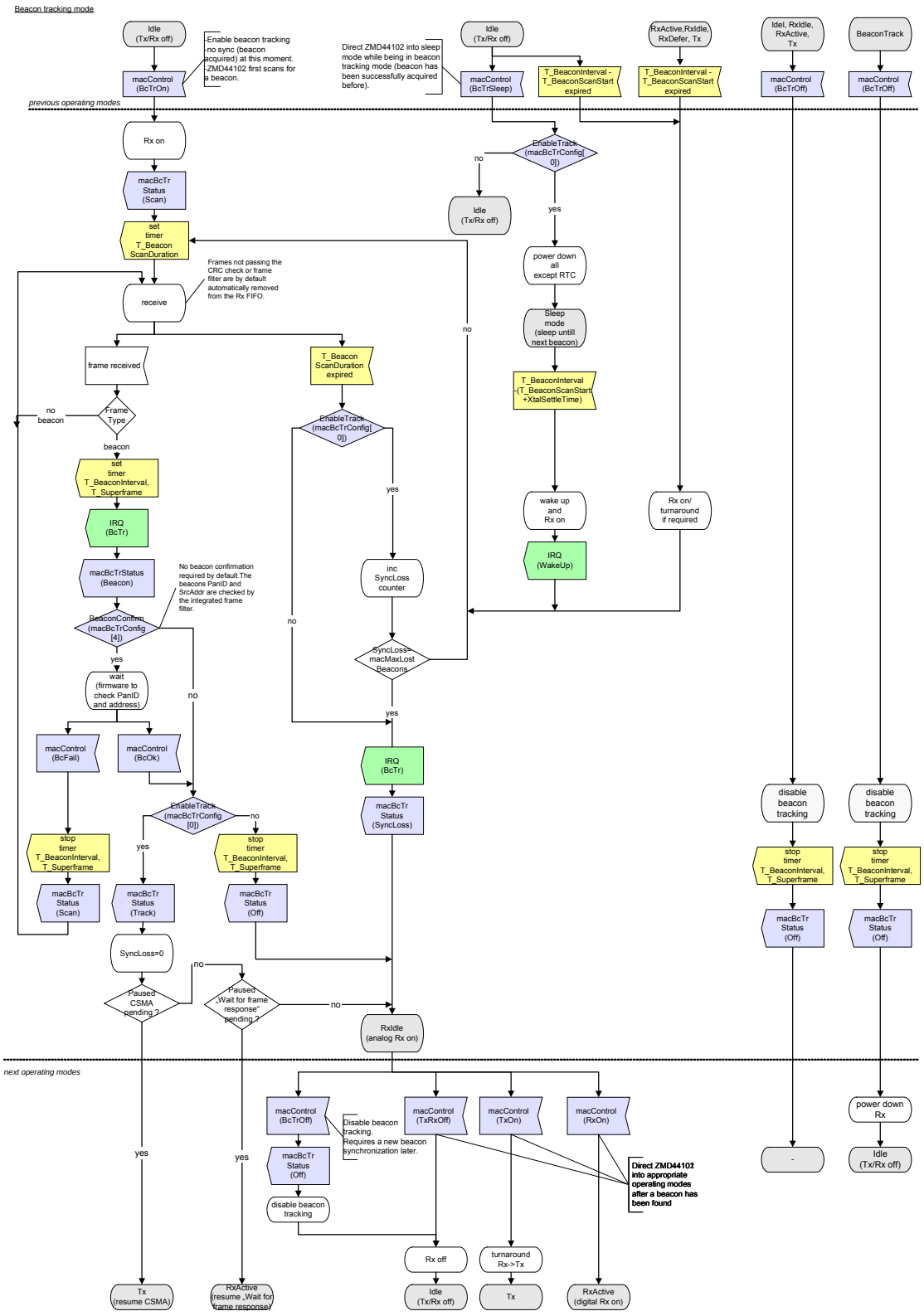


Figure 4.23: Beacon Tracking Mode SDL Diagram

Note that during the beacon tracking scan phase, the HW-MAC Rx function is used. Therefore, a successful data reception and a CRC failure are also indicated by an Rx and CRCfail interrupts. If these additional interrupts are not required, the user can ignore them in the IRQreason register.

4.11.7 Superframe Configuration

A superframe is used in a beacon-enabled network. The superframe is bound by the beacon frames transmitted by the network coordinator.

The registers used for the superframe configuration are the same as those for coordinator and slave devices. The beacon interval length must be set up in the T_BeaconInterval register. The duration of the active portion of the superframe (SD) is defined by the macSuperframeOrder (SO):

SD = 960 * 2^{SO} symbols

SD must be smaller than or equal to the beacon interval T_BeaconInterval. SO must be in the range from 0 to 14. Note that an SO equal to 15 is not allowed. In 802.15.4, the value 15 is used to disable the superframe. In the ZWIR4501, the EnableSlottedMode bit must be set to zero in the macTxConfig register to disable any slotted transmission.
Note that in beacon tracking mode, the T_BeaconInterval and macSuperframeOrder registers can be adjusted on-the-fly after each tracked beacon. This may be required to update the superframe configuration according to the superframe specification extracted from the payload of the tracked beacon frame.

The active portion of each superframe is divided into 16 slots, each of which is composed of a contention access period (CAP) and a contention free period (CFP). The CAP always commences immediately after the beacon frame and extends until the end of the slot defined in the macCAPend register. Note that the standard requires a minimum CAP length of 440 symbols. A transmission using slotted CSMA mode will be located within the CAP automatically by the ZWIR4501. A CAP transaction that cannot be completed before the CAP end is not executed and generates a failure status in the macTxStatus register.

The CFP can follow the CAP. Within the CFP, one or more guaranteed time slots (GTS) can be assigned. The GTS starts with the slot number macGTSstart and has a length of macGTSlength. This value must not exceed the number of slots from the beginning of the GTS to slot 15. If macGTSlength is set to 0, then no GTS is assigned and the macGTSstart value is ignored. If macGTSlength is greater than 0, macGTSstart must be greater than macCAPend. GTS transmissions are located within the GTS automatically by the ZWIR4501. A GTS transaction, which cannot be completed before the GTS end, is not executed and generates a failure status in the macTxStatus register.

For details on the Tx mode and the GTS/CAP transaction checks, refer to section 4.11.2. For details about the superframe structure requirements of the standard refer to *IEEE 802.15.4, Section 7.5.1.1.*

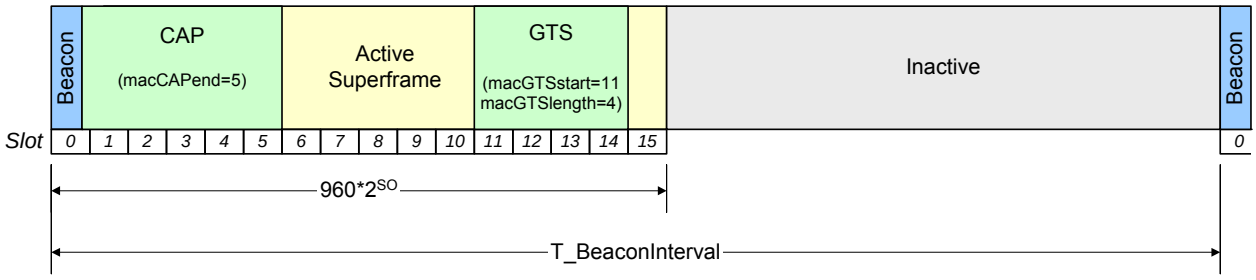


Figure 4.24: Superframe Configuration

4.12 Timing Correction

4.12.1 Network Timing Correction

In a beacon-enabled network, the coordinator and the beacon tracking device share the same superframe time base. The frequency deviation of the crystals leads to a timing error between the coordinator's and the device's time base. This timing error will be close to zero immediate following the beacon frame and will then increase over the beacon interval. The tolerable limit of the timing deviation for a coordinator defined by IEEE_standard is 40 ppm. The ZWIR4501 uses the 32.768 kHz crystal based RTC for the beacon tracking time base. The worst case frequency deviation for a low cost 32.768 kHz crystal can be assumed to be 80 ppm. Hence, the total beacon interval timing deviation between coordinator and a device can be up to 40 ppm + 80 ppm =120 ppm.
Table 4.1 shows the timing deviation at the end of the beacon interval for 120 ppm and different beacon order.

Beacon Order	Beacon Interval	Timing Deviation (120ppm)	
	in symbols	in symbols	in ms (channel 0)
6	61440	7	0,18
8	245760	29	0,74
12	3932160	472	11,80
14	15728640	1.887	47,19

Table 4.1: Timing Deviation for 120ppm

In beacon tracking mode, the ZWIR4501 can track and correct the beacon interval timing error relative to its coordinator. This minimizes the time the receiver must be turned on before the arrival of the next beacon and therefore helps reduce the power consumption. Additionally, the superframe time is periodically corrected in order to keep the slot boundaries aligned with the coordinators' slot boundaries, which reduces collisions in the slotted traffic. Moreover, a beacon tracking device can go to sleep within the superframe and recover the superframe time on wake up, allowing a device, for example, to sleep until a GTS transmission.
The features mentioned above will be explained in subsequent Sections.

4.12.2 Beacon Interval Timing Correction

The integrated beacon interval timing correction measures the time difference between the expected and the actual arrival of the beacon frame. The difference is used to adjust the beacon interval and to improve the estimate of the next beacon arrival. Figure 4.25: Beacon Tracking Timing Correction illustrates this process. The T_BeaconScanStart register is used to control the time the receiver is turned on before the expected arrival of the next beacon frame.
At the end of the initial beacon interval, the uncertainty window of the beacon arrival is two times the worst case timing deviation, to which the T_BeaconScanStart register must be adjusted.

Given a total coordinator-device timing deviation (DEV) of 120ppm and a beacon order (BO), the initial T_BeaconScanStart time is:

$$T\_BeaconScanStart = 960*2^{BO}*DEV$$
$$T\_BeaconScanStart = 960*2^{BO}*120e\text{-}6$$

During the initial beacon interval, the macBcTrStatus shows the status Track and Sync, where Sync means that the HW-MAC has successfully synchronized to a beacon but the beacon interval is not corrected yet. After the 2nd tracked beacon, the beacon interval timing correction is active, as indicated by the additional Align status bit in the macBcTrStatus register. At this time, the beacon arrival uncertainty window should be reduced by setting T_BeaconScanStart to 5. This will reduce the beacon tracking power consumption.

Whenever a beacon frame is lost, the beacon interval timing correction requires the addition of one beacon interval in order to realign with the coordinator's beacon interval. During this time, the macBcTrStatus Align bit is cleared and only the Sync indicator bit is set. The T_BeaconScanStart must be reset to the worst case timing deviation.

In general, after each tracked beacon, the firmware should check the Align bit in the macBcTrStatus register. If the bit is set, the T_BeaconScanStart should be set to 5, otherwise, it should be set to the worst case timing deviation.

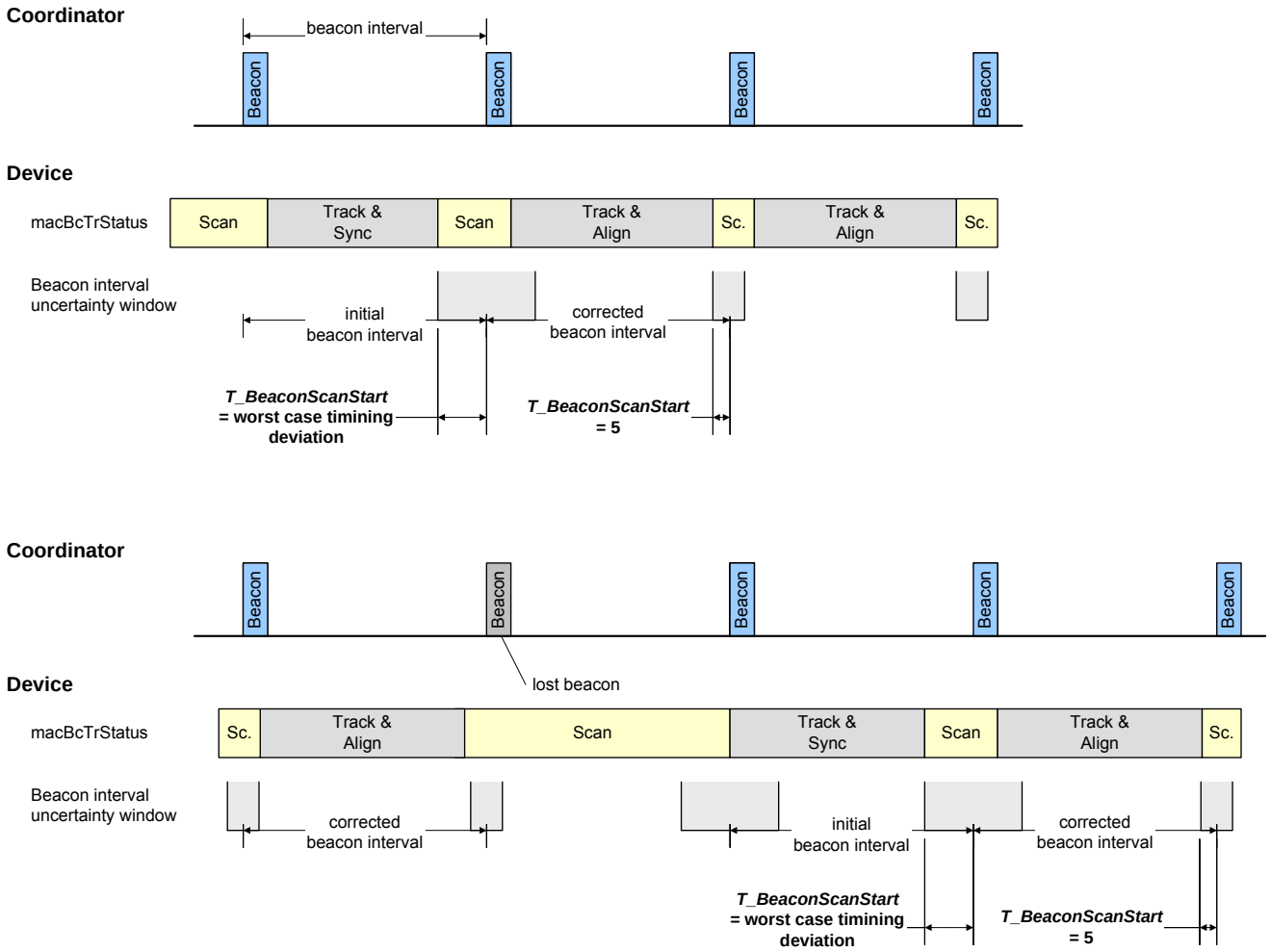


Figure 4.25: Beacon Tracking Timing Correction

4.12.3 Superframe Timing Correction

There is also symbol time drift within the superframe between a beacon tracking device and a coordinator. The minimum symbol time accuracy according to the IEEE_standard is 40 ppm; therefore the worst case symbol time deviation between a coordinator and a device is 40 ppm + 40 ppm = 80 ppm.
The ZWIR4501 aligns its beacon tracking interval to the beacon interval of the coordinator. Once a beacon tracking device has its beacon interval aligned (Align bit set in the macBcTrStatus register), it will automatically correct its superframe time base. The correction interval is adjusted in the SFalignOrder register. The correction interval should not be too short, since there will be a residual error of <5 symbols caused by the automatic superframe time correction algorithm.

Given a timing deviation (DEV) in ppm, the SFalignOrder should be set to:

SFalignOrder > $\log_2(5/60/DEV)$

For a total symbol time deviation of 80 ppm for example, the SFalignOrder should be adjusted to:

SFalignOrder >= $\log_2(5/60/80e\text{-}6)$

SFalignOrder = 11

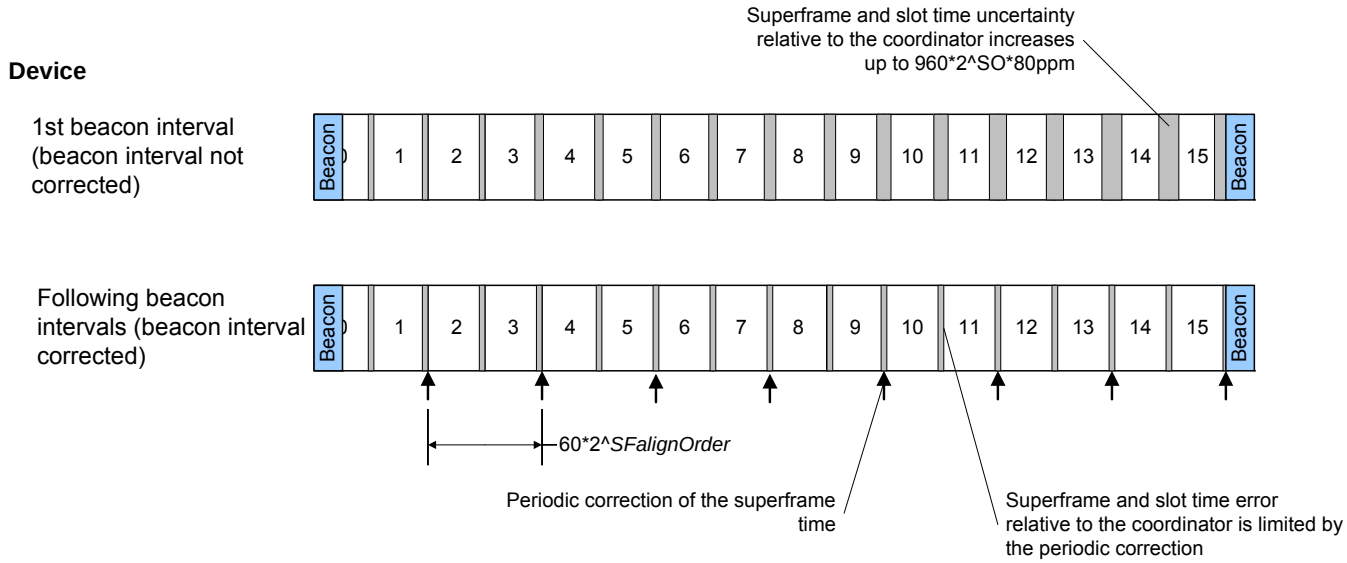


Figure 4.26: Superframe Timing Correction

Since the superframe timing is not corrected during the initial beacon interval and superframe timing correction cannot completely eliminate the timing error, a guard time should be added to the slotted traffic by the HW-MAC in order to avoid collisions.
There is both an early and a late guard time that can be enabled in the macBcTrConfig register. The early guard time is used to delay the GTS slot start. The late guard time is added to transaction length for the CAP and GTS end checks.

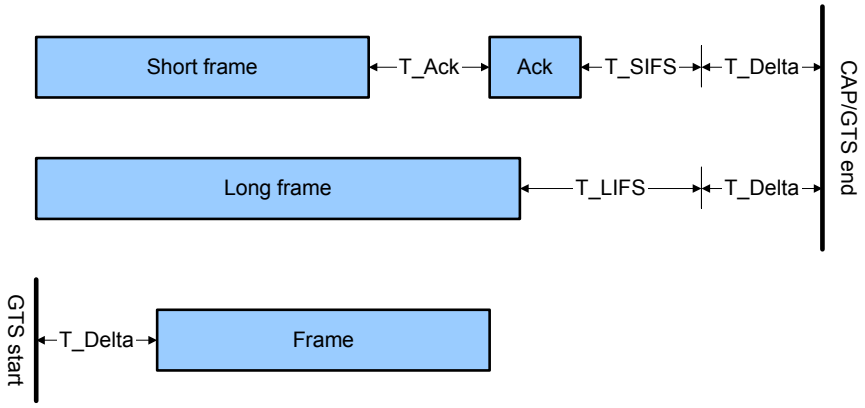


Figure 4.27: Early and Late Guard Time T_Delta

The required guard time length T_Delta depends upon the beacon interval alignment status. During an initial beacon interval (macBcTrStatus Align bit is not set), and depending upon the superframe order (SO) and the worst case symbol timing deviation (DEV) between the coordinator and a device, where the superframe timing correction is not active, the guard time should be set:

T_Delta = $960*2^{SO}$*DEV

Once the beacon interval is aligned to coordinator beacon interval (macBcTrStatus Align bit is set) and the periodic superframe timing correction is active, the guard time can be reduced to:

T_Delta = MIN(5, T_Delta = $960*2^{SO}$*DEV)

symbols.

4.13 General Purpose Timer Function

4.13.1 Overview

The HW-MAC provides 3 different timers which can be used for the following functions:

1. Program a timer-controlled general purpose interrupt
2. Perform a timer-controlled execution of a MAC control command
3. Generate timestamps for receive and transmit frames

The 3 available timers have already been introduced in sections 4.11.5 and 4.11.6.

1. T_TotalTimeFFD
   This timer can either be initiated by the macControl command TotalTimerStart or if not running it will be started together with the automatic beacon transmitting function (AutoBeaconTx). It is 24-bit wide up-counting timer with symbol resolution, which is driven by the 24 MHz crystal. Therefore this timer will be paused during Sleep or Gobal Power Down mode. The current TotalTimeFFD timer value can be read from the macTotalTimeFFD register.

2. T_TotalTimeRFD
   This timer is only available in beacon tracking mode. It is driven by the 32.768 kHz crystal and has a resolution of RTC units (1/(32.768 kHz) in EU mode and 2/(32.768 kHz) in US mode). It is a count down timer, counting from the arrival of the last beacon down to 0, which is the expected arrival of the next beacon. The TotalTimeRFD timer is also running during Sleep mode. The current TotalTimeRFD timer value can be read from the macTotalTimeRFD register.

3. T_Superframe
   This timer counts the symbols of the active portion of the superframe starting from 0 at the beginning of the superframe. It is only available after each transmitted or tracked beacon frame. Its content can be monitored from the macCurrentSymbolTime register.

These 3 timers can be selected independently as timer sources for general purpose interrupt and timer-controlled MAC control functions and for the receive and transmit timestamp generation. The timer selection is controlled by the macTimerConfig register.

4.13.2 General Purpose Interrupt

The general purpose interrupt is a free programmable timer controlled interrupt. First a running timer must be selected in the macTimerConfig register. The time of the general purpose interrupt must be programmed in the 24-bit T_General register. The general purpose timer function is activated by the macControl command GeneralTimerStart. When the selected timer reaches the value programmed in T_General, an interrupt with the IRQreason Timer is generated. The macTimerControlStatus register will show the status GeneralExp.

4.13.3 Timer Controlled MAC Control Command Execution

The macControlT command execution timer can be used to run a timer-controlled execution of a MAC control command. First a timer must be selected in the macTimerConfig register. If the device does not run in the beacon transmit mode, the timer must be initiated by the macControl command TotalTimerStart. The MAC control execution time (absolute time value) must be programmed in the 24-bit T_MacControl register. This timer function is activated by writing a MAC control command to the macControlT register. When the selected timer reaches the value programmed in T_MacControl register an interrupt with the IRQreason Timer is generated and the command in the macControlT register is released to the HW-MAC. The macTimerControlStatus register will show the status macControlExp.
The MAC control command execution and rejection indication is the same as described in section 4.3.

4.13.4 Receive and Transmit Timestamp

The integrated HW-MAC generates a timestamp for each received and transmitted frame.
The receive timestamp is taken at the end of the SFD and stored in the macRxTime register. The transmit timestamp is taken at the leading edge of the PHY frame and stored in the macTxTime register. For tracked and generated beacon frames, the timestamp is also copied to the macBeaconRxTime and macBeaconTxTime register. The timer source for the timestamp generation is selected in the macTimerConfig register.

4.14 Message Sequence Charts (MSCs)

The following message sequence charts demonstrate how to control the ZWIR4501 by a microcontroller in different typical scenarios. The MSCs are not intended to be standard compliant. The communication scenarios shown can be implemented in different ways. Only a typical solution is depicted. See section 4.11 for detailed description on the different operation modes.
The ZWIR4501 can be used with a software implementation (like [9]) to provide an IEEE 802.15.4 compliant MAC interface.

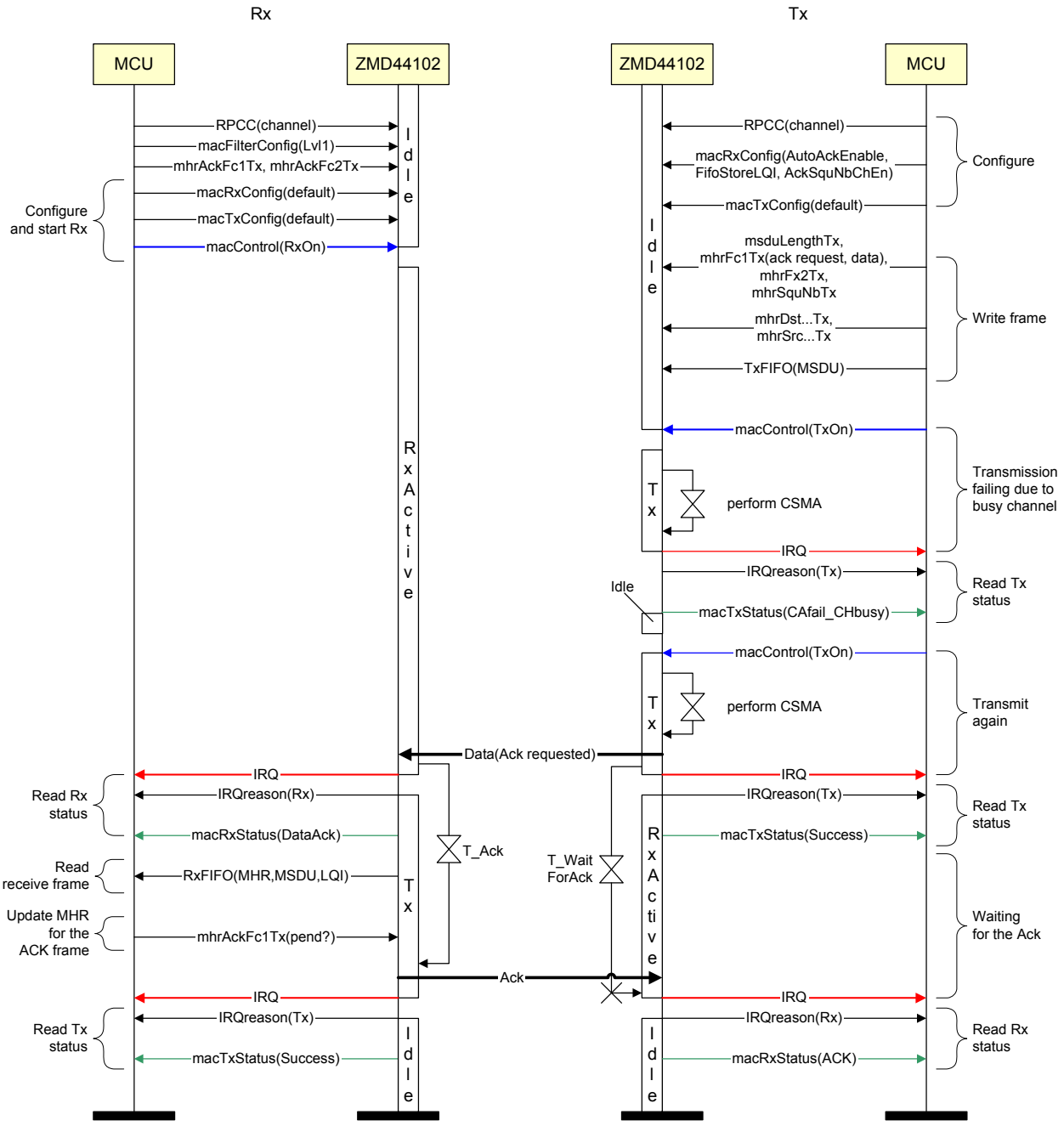


Figure 4.28: MSC Unslotted Tx/Rx with Automatic Ack

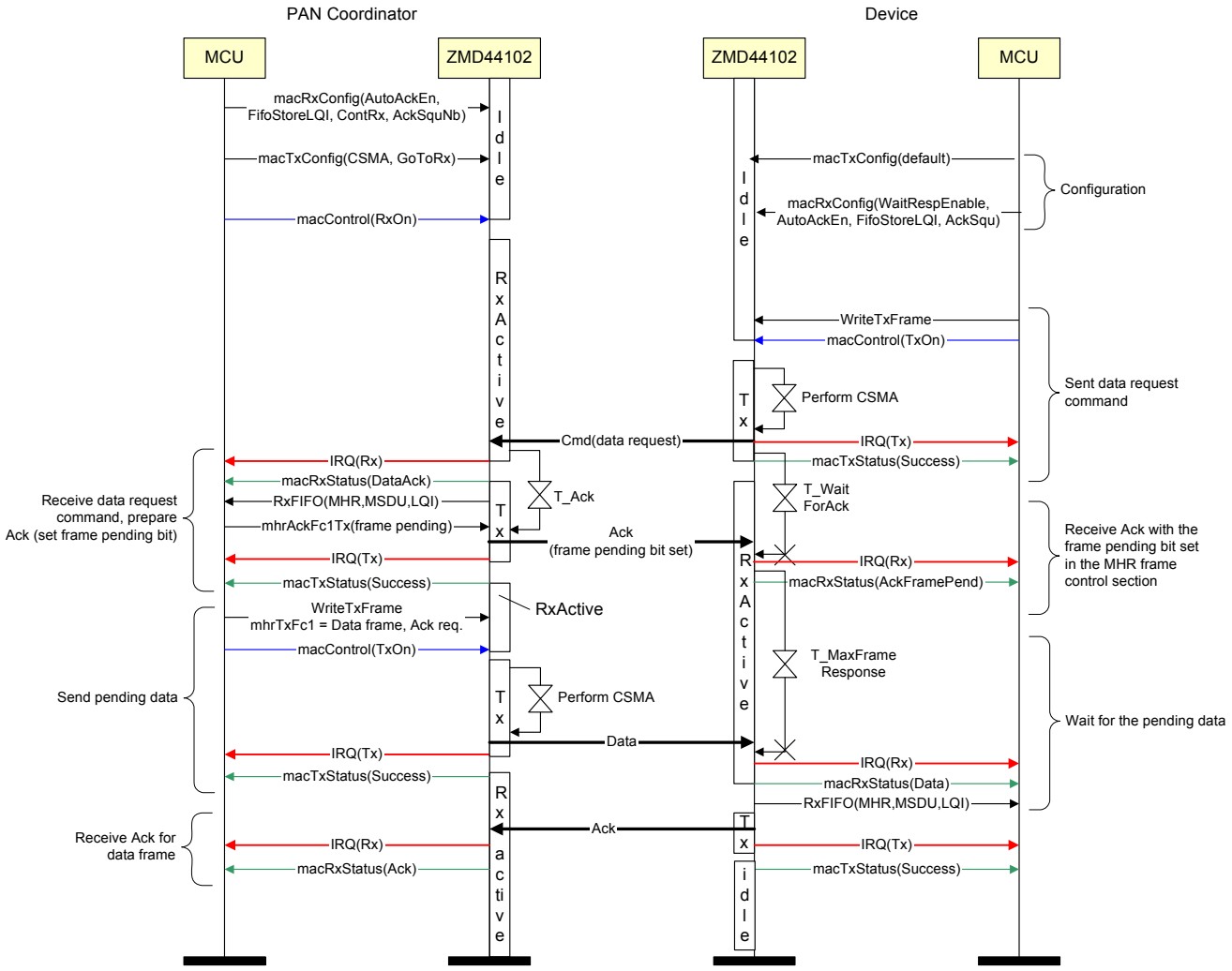


Figure 4.29: MSC Unslotted Data Polling

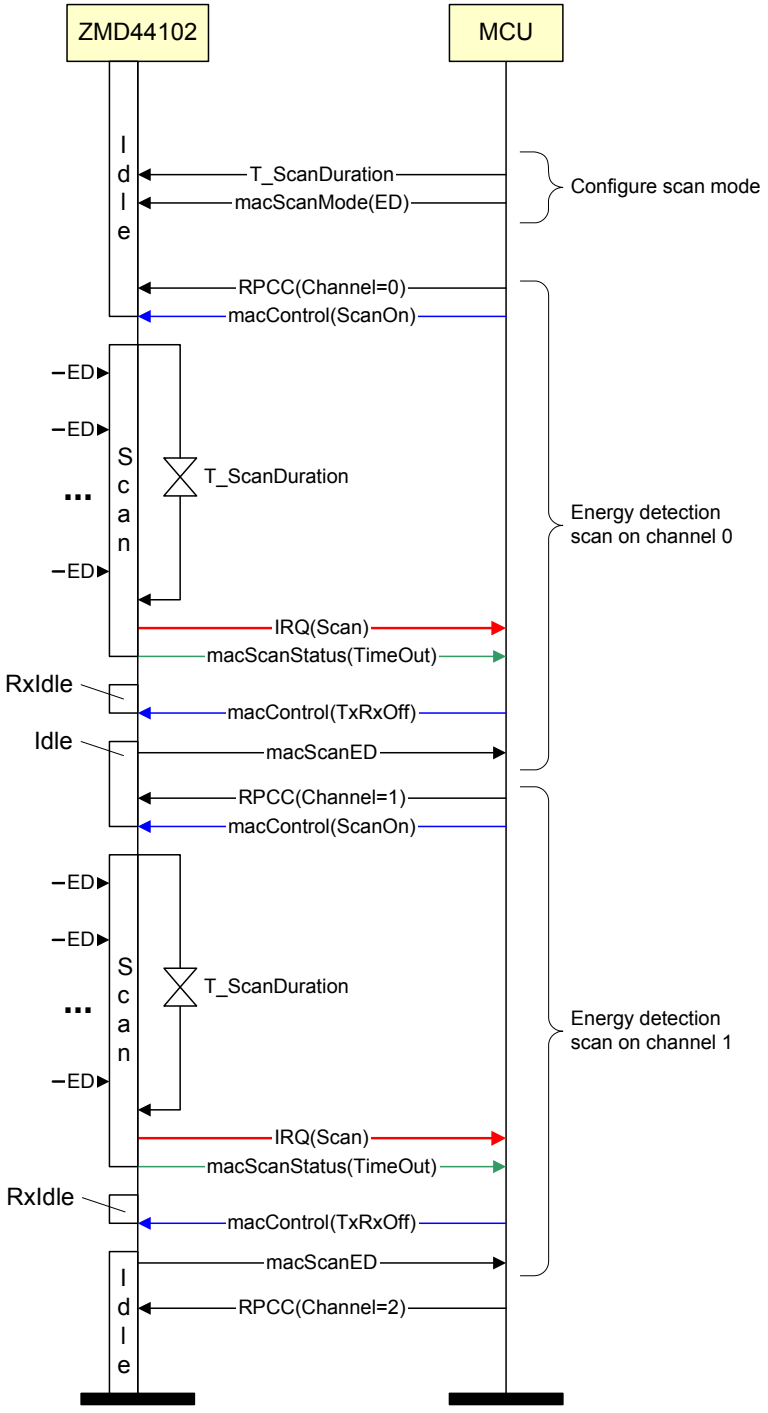


Figure 4.30: MSC Energy Detection Scan

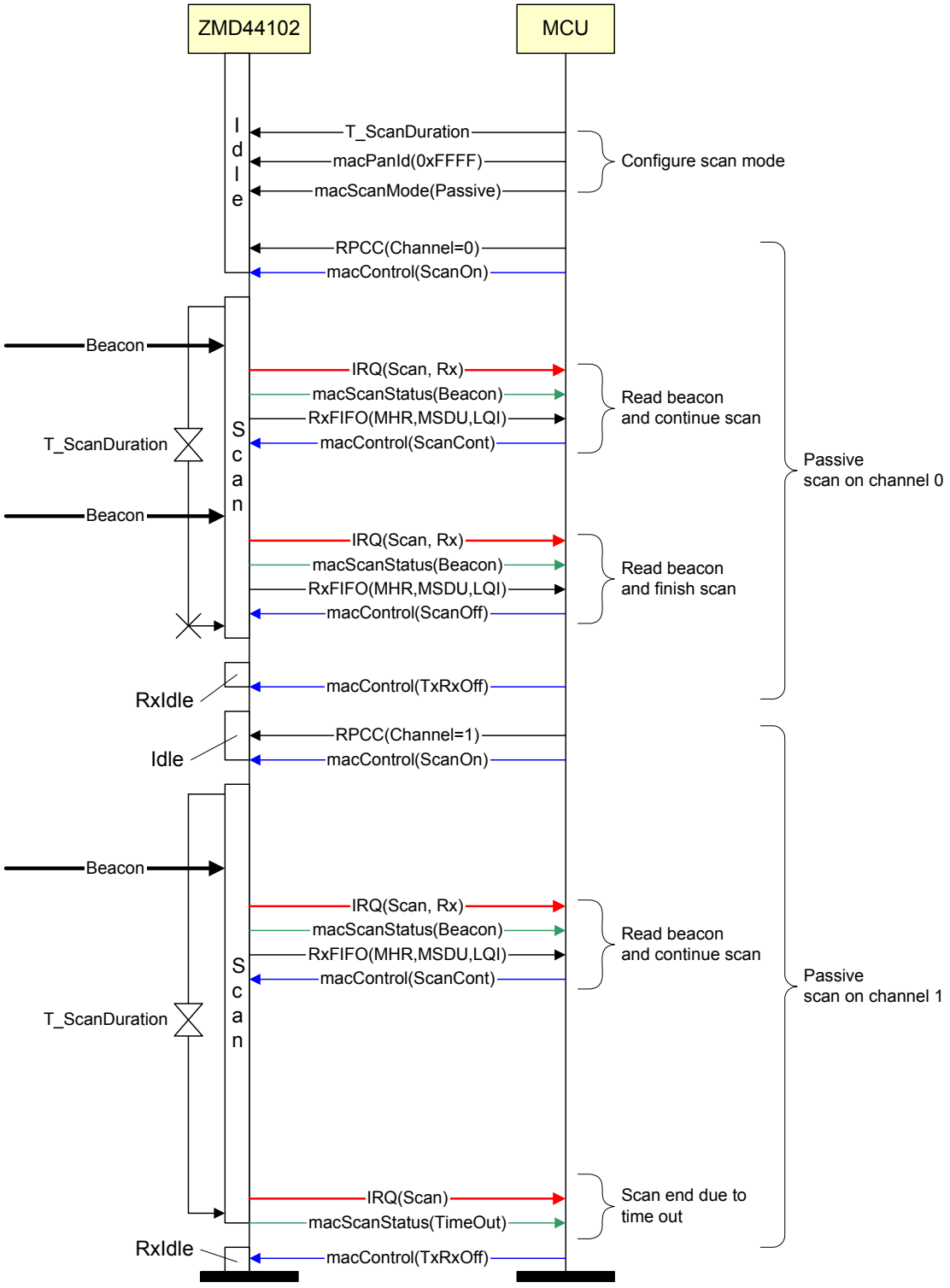


Figure 4.31: MSC Passive Scan

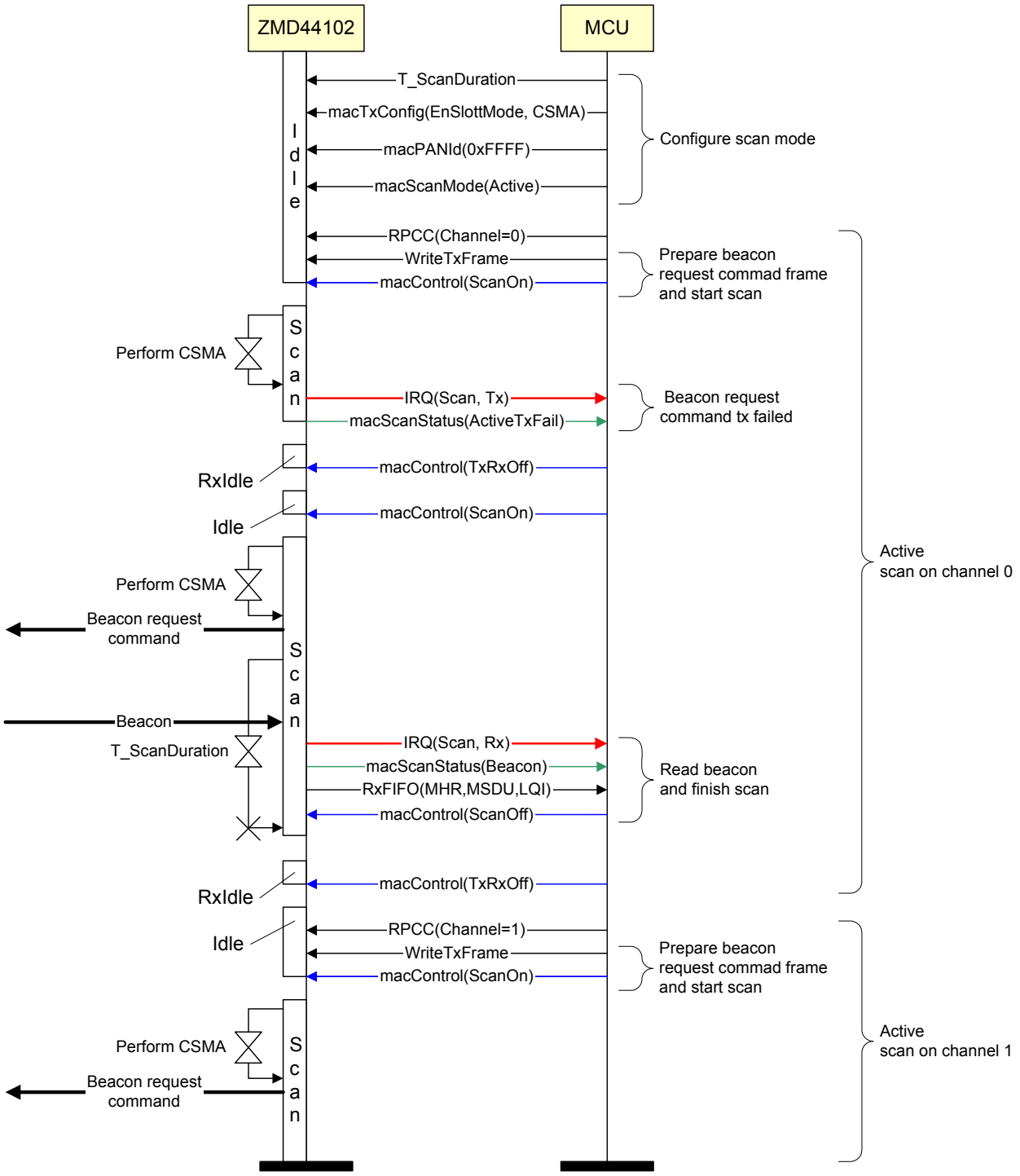


Figure 4.32: MSC Active Scan

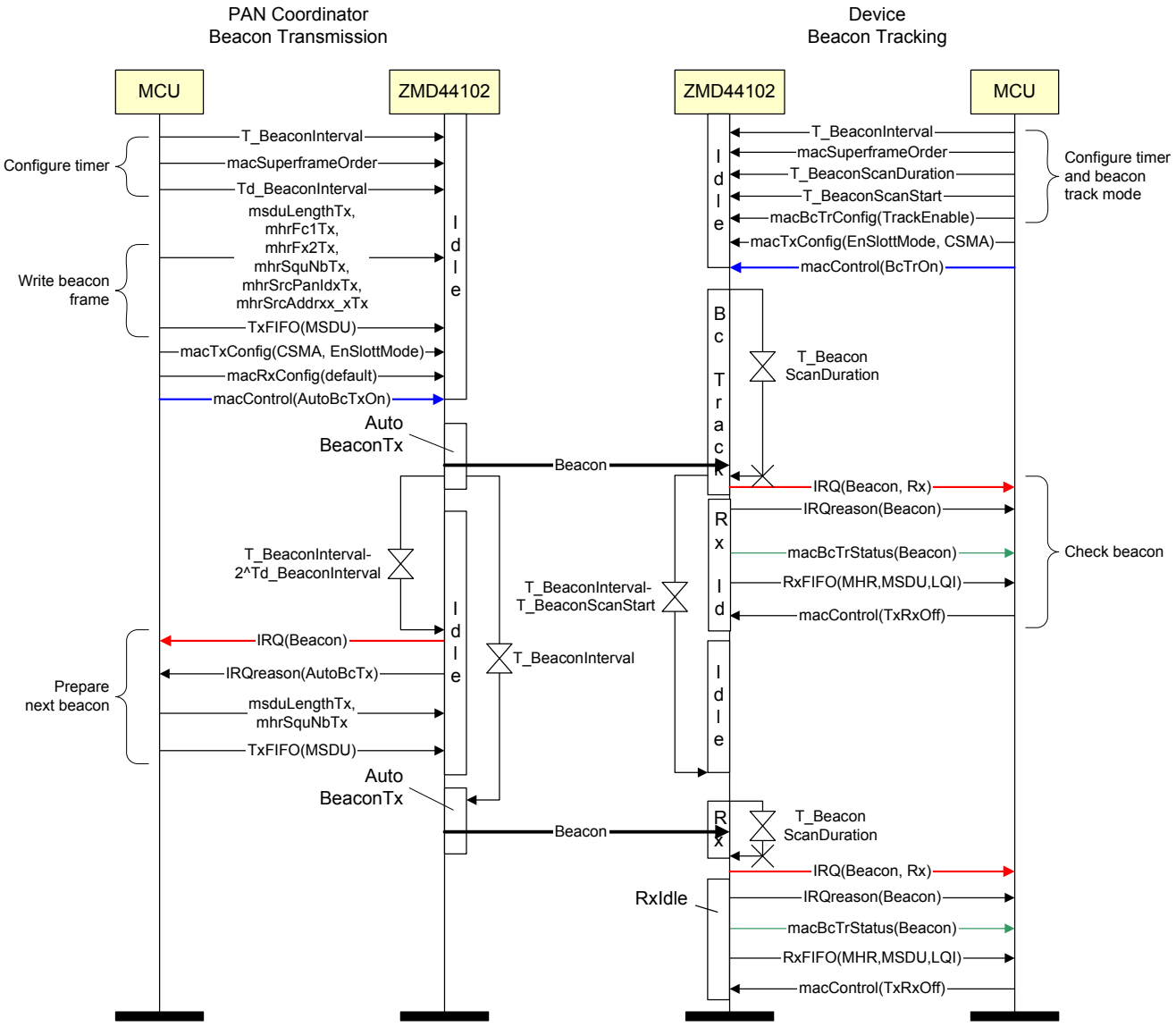


Figure 4.33: MSC Basic Beacon Transmission and Tracking

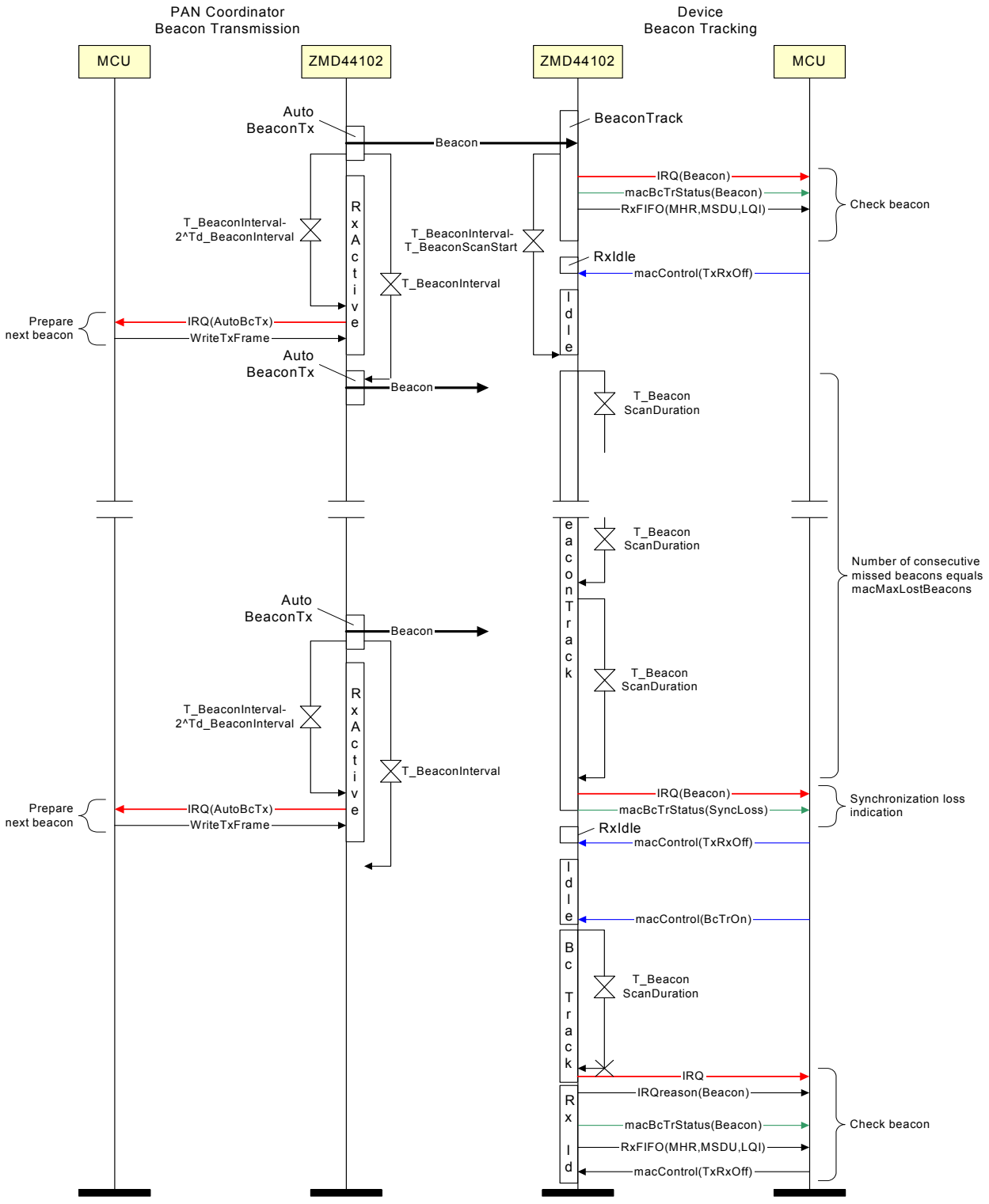


Figure 4.34: MSC Beaconing Synchronization Loss

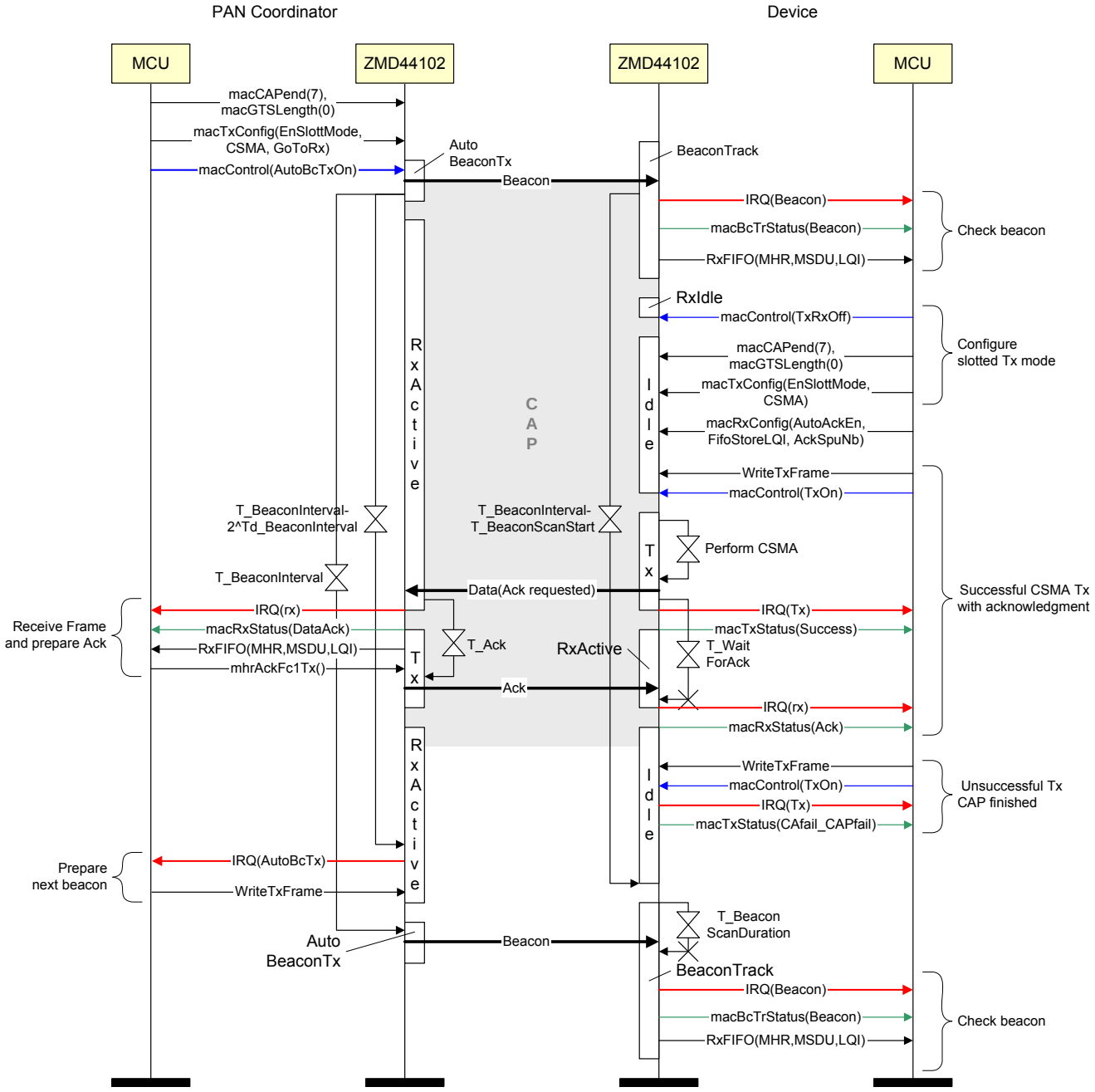


Figure 4.35: MSC Slotted Tx/Rx in CAP

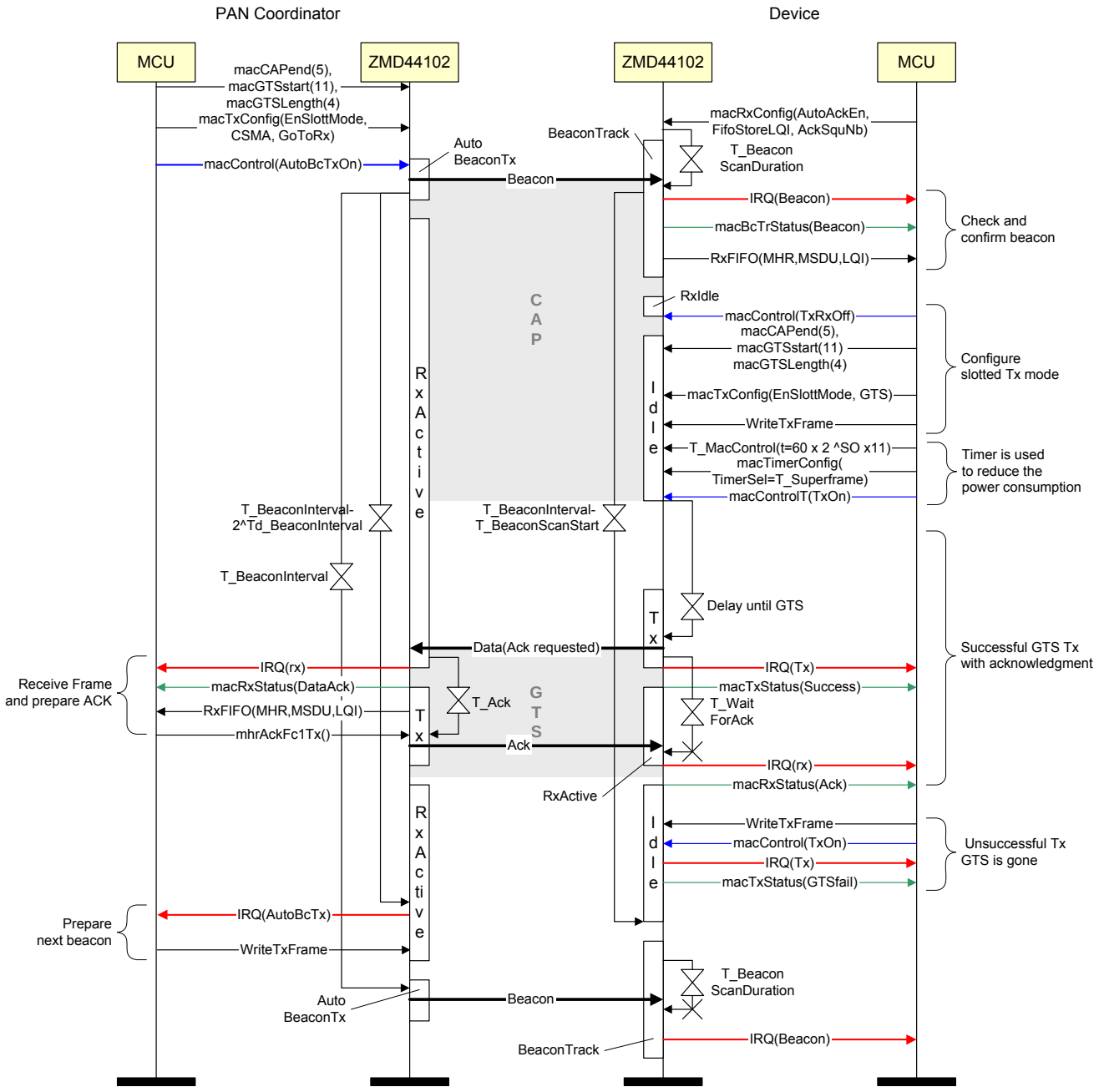


Figure 4.36: MSC Slotted Tx/Rx in GTS

In this example, a superframe configuration is used as it is shown in Figure 4.24, and the device transmits its data in the first slot of its GTS (slot #11).

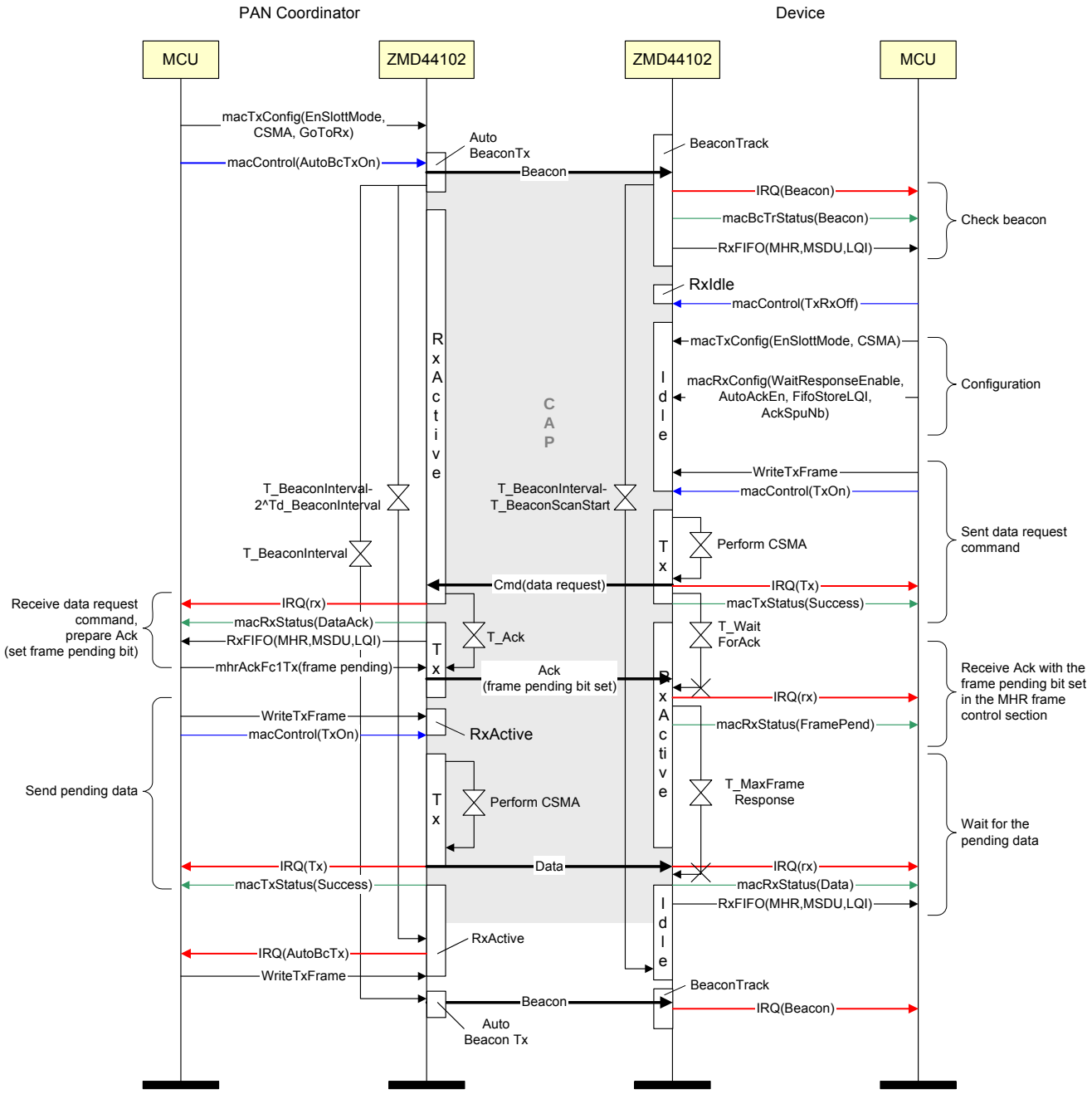


Figure 4.37: MSC Extracting Data from the Coordinator

5 Registers

This chapter contains descriptions for the ZWIR4501's registers. These registers can be modified through the SPI or parallel interface by utilizing an additional microcontroller. The register description contains a mnemonic for easy referencing, a bit description where applicable, the access mode (R: read and/or W: write) and the address of the register within the ZWIR4501.
ZMD provides a C code header file that provides the C code-based constant definitions for the register's address and other C code-based constant definitions as commands, status and configuration settings [6]. When starting to write C code for any MCU that controls the ZWIR4501, contact ZMD to receive an up-to-date version of this header file. The C code-based constant address definition is noted with the register description for easier handling.
The register description includes a "Reset" column, which provides information about the initial register's content after the reset procedure.

5.1 Register Summary

IRQ Control Register

Mnemonic	Address	Short Description	See page
IRQreason	0xAC	Interrupt Request Reason	75
IRQmask1, 2, 3, 4	0xAD – 0xB0	Interrupt Condition Mask 1, 2, 3, 4	77

PHY Register

Mnemonic	Address	Short Description	See page
RPCC	0x00	PHY Current Channel	78
RTXM	0x05	Transmit Mode Register	79
RxMode	0x08	Receiver Mode Register	79
EdThreshold	0x24	Energy Detection Threshold	79
AgcLvl	0x0E	Automatic Gain Control Level	80

MAC Operating Control

Mnemonic	Address	Short Description	See page
macControl	0xA0	MAC Control Commands	81
macControlT	0xA1	MAC Control Commands, Timer-Controlled	81
macOpMode	0xA2	MAC Operating Mode	82
RPD	0x14	Power Down Control Register	83

MAC FIFO Register

Mnemonic	Address	Short Description	See page
TxFIFO	0x80	Transmit FIFO	83
RxFIFO	0x81	Receive FIFO	83
macFifoStatus	0xA9	MAC Tx/Rx FIFO Status	84
macFramePend	0x88	Frame Pending Number	84

MAC Tx Control

Mnemonic	Address	Short Description	See page
macTxConfig	0xB1	MAC Transmitter Configuration	85
macTxStatus	0xA3	MAC Transmit Status	86
msduLengthTx	0x60	Transmit Frame MSDU Length	86
mhrFc1Tx	0x61	Transmit Frame MAC Header Frame Control Field 1	87
mhrFc2Tx	0x62	Transmit Frame MAC Header Frame Control Field 2	87
mhrSquNbTx	0x63	Transmit Frame MAC Header Sequence Number Field	88
mhrDstPanIdTx	0x64 – 0x65	Tx Frame MAC Header Dest. PAN Identifier Field	88
mhrDstAddr16	0x66 – 0x67	Tx Frame MAC Header 16-bit Dest. Address Field	88
mhrDstAddr64Tx	0x68 – 0x6F	Tx Frame MAC Header 64-bit Dest. Address Field	89
mhrSrcPanIdTx	0x70 – 0x71	Transmit Frame MAC Header Source PAN Identifier Field	89
mhrSrcAddr16Tx	0x72 – 0x73	Transmit Frame MAC Header 16-bit Source Address Field	89
mhrSrcAddr64Tx	0x74 – 0x7B	Transmit Frame MAC Header 64-bit Source Address Field	90
mfrCRCTx	0x7C – 0x7D	Transmit Frame CRC Field	90
macTxTime	0xF8 – 0xFA	Transmit Timestamp	90

MAC Rx Control

Mnemonic	Address	Short Description	See page
macRxConfig	0xB2	MAC Receiver Configuration	91
macRxStatus	0xA4	MAC Receive Status	92
mhrFcRx	0x82 – 0x83	Received Frame MAC Header Frame Control Field	93
mhrSquNbRx	0x84	Received Frame MAC Header Sequence Number	93
mpduLengthRx	0x85	Received Frame MPDU Length	93
mfrCRCRx	0x86 – 0x87	Received Frame CRC Field	93
macRxTime	0xF5 – 0xF7	Receive Timestamp	94

MAC Ack Control

Mnemonic	Address	Short Description	See page
mhrAckFc1Tx	0x7E	Transmit Ack Frame MAC Header Frame Control Field	94
mhrAckFc2Tx	0x7F	Transmit Ack Frame MAC Header Frame Control Field	95
T_Ack	0xCC	Frame to Acknowledge Space	95
T_WaitForAck	0xCF	Wait for Acknowledge Time	95
T_MaxFrameResponse	0xD0 – 0xD1	Maximum Frame Response Wait Time	96

MAC Scan Control

Mnemonic	Address	Short Description	See page
macScanStatus	0xA5	MAC Scan Status	97
macScanED	0xA6	Maximum Measured ED-Scan Energy	97
macScanMode	0xB3	MAC Scan Mode Configuration	98
T_ScanDuration	0xD2 – 0xD4	Scan Duration Time	98

MAC Beacon Control

Mnemonic	Address	Short Description	See page
T_BeaconInterval	0xD5 – 0xD7	Beacon Interval	99
macBcTxConfig	0xB5	MAC Automatic Beacon Transmit Configuration	99
macAutoBcTxStatus	0xA8	MAC Automatic Beacon Transmission Status	99
Td_BeaconInterval	0xD8	Beacon Interval Delta Time Exponent	100
macBeaconTxTime	0xF2 – 0xF4	Beacon Transmit Timestamp	100
macBcTrConfig	0xB4	MAC Beacon Tracking Configuration	101
macBcTrStatus	0xA7	MAC Beacon Track Status	102
macCoordSrcAddr	0x95 – 0x96	Coordinator Short Address	103
macCoordExtAddr	0x97 – 0x9E	Coordinator Extended Address	103
macMaxLostBeacons	0xC3	Maximum Lost Beacon Number	103
macSyncLoss	0xC4	Current Number of Lost Beacon Number	104
BcTrThreshold	0xC7	Beacon Tracking Timing Correction Threshold	104
T_BeaconScanDuration	0xD9 – 0xDB	Beacon Scan Duration Time	104
T_BeaconScanStart	0xDC – 0xDD	Beacon Scan Start Time	105
T_Delta	0xE1 – 0xE2	Timing Error Guard Time	105
macBeaconTrackError	0xED – 0xEE	Last Received Beacon Tracking Error	105
macBeaconRxTime	0xEF – 0xF1	Beacon Receive Timestamp	106

MAC Timer Control and Value

Mnemonic	Address	Short Description	See page
macTimerControlStatus	0xAB	MAC Timer and Control Status	106
macTimerConfig	0xB7	MAC General Purpose Timer Configuration	106
T_MacControl	0xC9 – 0xCB	macControlT Command Execution Time	107
T_General	0xDE – 0xE0	General Purpose Timer and Sleep Time	107
macTotalTimeFFD	0xE3 – 0xE5	Current FFD Network Time	108
macTotalTimeRFD	0xE6 – 0xE8	Current RFD Network Time	108

MAC Frame Filter Control

Mnemonic	Address	Short Description	See page
macFilterConfig	0xB6	MAC Frame Filter Configuration	109
macFilterStatus	0xAA	MAC Frame Filter Status	109
macPanId	0x89 – 0x8A	Reference MAC PAN Identifier	110
macShortAddr	0x8B – 0x8C	Reference MAC Short Address	110
aExtendedAddr	0x8D – 0x94	aExtendedAddr – Reference Extended Address	110

MAC Superframe and GTS Control

Mnemonic	Address	Short Description	See page
macSuperframeOrder	0xBF	Superframe Order (SO)	111
macCAPend	0xC0	Contention Access Period (CAP) End	111
macGTSstart	0xC1	Guaranteed Time Slot (GTS) Start	111
macGTSlength	0xC2	Guaranteed Time Slot (GTS) Length	112
macCurrentSymbolTime	0xE9 – 0xEB	Current Symbol Time	112
macCurrentSlot	0xEC	Current Slot Number	112
macMaxSIFSFrameSize	0xC5	Maximum Short Interframe Space (SIFS) Frame Size	112
SFalignOrder	0xC6	Superframe Alignment Order	113
T_SIFS	0xCD	Short Inter-Frame Spacing	113
T_LIFS	0xCE	Long Inter-Frame Spacing	113

MAC CSMA Control

Mnemonic	Address	Short Description	See page
macUnitBackOffPeriod	0xC8	Unit Backoff Period	114
macMinBe	0xB9	Minimum CSMA-CA Backoff Exponent	114
macMaxBe	0xBA	Maximum CSMA-CA Backoff Exponent	114
macInitialCW	0xBB	CSMA_CA Initial Contention Window Size	114
macMaxCSMABackOffs	0xBC	Maximal Number of CSMA-CA Backoffs	115
macBattLifeExtPeriods	0xBD	Battery Life Extension Window Length	115
CsmaSeed	0xBE	CSMA Random Backoff Generator Seed	115

SPI Registers

Mnemonic	Address	Short Description	See page
SPIconfig	0xFB	SPI Configuration	116
SPIstart	0xFC	SPI Master Mode Start	117
SPItx	0xFD	SPI Master Mode Transmit Byte	117
SPIrx	0xFE	SPI Master Mode Receive Byte	117

Clock Output Configuration

Mnemonic	Address	Short Description	See page
ClkoutConfig	0xB8	CLKO Output Configuration	118

Recommended Default Register Setup

Mnemonic	Address	Short Description	See page
MTPcontrol	0x37	MTP Control Register	119

5.2 IRQ Control, Configuration and Status

Register: Interrupt Request Reason [7:0]
mnemonic IRQreason, address 0xAC, access: R, C code-based constant address definition: IRQ_REASON

Addr.	Register	Bit	Description	Reset
0xAC	IRQreason	0	WakeUp, Timer, CmdError	0
		1	Rx	0
		2	Tx	0
		3	Scan	0
		4	BcTr	0
		5	AutoBcTx	0
		6	FIFO	0
		7	CRC/Frame failure	0

The reasons for pending interrupts are indicated by the high bits in the IRQreason register. After the MCU's software has read the IRQreason register, it is automatically cleared to 0x00 by hardware. After an interrupt has been issued further information on the actual status can be obtained by the corresponding status register, like macTxStatus or macRxStatus.

Each of the eight IRQ reasons can have different interrupt conditions:

Reason	Condition	Description	Associated Status Register
Wake Up/ Timer/ CmdError	WakeUp	Wake up from Sleep mode	N/A
	Timer	T_MacControl or T_General timer expiration	macTimerControl Status
	CmdError	Wrong macControl or macControlT command	
Rx	Ack	Acknowledge frame received	macRxStatus
	AckTimeOut	No Ack frame received within T_WaitForAck	
	AckFramePend	Acknowledge frame received with frame pending bit set	
	PollNoData	No pending data received within T_MaxFrameResponse	
	PollNoCAPend	No pending data received within the current CAP. WaitForFrame process will resume in the next CAP.	
	Data	Data frame without Ack requested received	
	DataAck	Data frame with Ack requested received	
Tx	Success	Transmission successfully finished	macTxStatus
	CAfail_CHbusy	Channel access failure due to busy channel	
	CAfail_CAPfail	Channel access failure because CAP end reached	
	GTSfail	GTS transmission failure because GTS has gone	
	CAfail_SFend	Channel access failure due to superframe end	
Scan	Command	Orphan scan: command frame received	macScanStatus
	Beacon	Active/passive scan: beacon frame received	
	TimeOut	Scan duration timed out	
	ActiveTxFail	Active scan: command transmission failed	
	OrphanTxFail	Orphan scan: command transmission failed	
BcTr	SyncLoss	Number of macMaxLostBeacons beacons lost during beacon tracking	macBcTrStatus
	Beacon	Beacon frame received during beacon tracking	
	CoordAddrFail	Beacon frame with received source address not matching the coordinator address	
Auto BcTx	TxReady	Indicate to firmware to prepare the next beacon	macAuto BcTxStatus
	TxDone	Beacon transmission is finished	
FIFO	TxUnderflow	Read access from the empty TxFIFO	macFifoStatus
	TxOverflow	Write access to the full TxFIFO	
	RxUnderflow	Read access from the empty RxFIFO	
	RxOverflow	Write access to the full RxFIFO	
CRC/ Frame fail	CRCfail	Frame with CRC failure received	macFilterStatus
	FrameFail	Frame received which failed the Address/FrameType check	

Register: Interrupt Condition Mask [30:0]
mnemonic IRQmask, address 0xAD – 0xB0, access: R/W, C code-based constant address definition: IRQ_MASK_1, IRQ_MASK_2, IRQ_MASK_3, IRQ_MASK_4

Address	Register	Bit	Reason	Condition	Reset
0xAD	IRQmask1	0	Wake Up	WakeUp	0
		1	Rx	Rx	1
		2	Rx	Ack	0
		3	Rx	AckTimeOut	0
		4	Rx	AckFramePend	0
		5	Rx	PollNoData	0
		6	Rx	PollCAPend	0
		7	Rx	Data	0
0xAE	IRQmask2	0	Rx	DataAck	0
		1	Tx	Success	0
		2	Tx	CAfail_CHbusy	0
		3	Tx	CAfail_CAP _fail	0
		4	Tx	GTSfail	0
		5	Tx	CAfail_SFend	0
		6	Scan	Command	0
		7	Scan	Beacon	0
0xAF	IRQmask3	0	Scan	TimeOut	0
		1	Scan	ActiveTxFail	0
		2	Scan	OrphanTxFail	0
		3	BcTr	SyncLoss	0
		4	BcTr	Beacon	0
		5	BcTr	CoordAdrFail	0
		6	AutoBcTx	TxReady	0
		7	AutoBTx	TxDone	1
0xB0	IRQmask4	0	FIFO	TxUnderflow	0
		1	FIFO	TxOverflow	0
		2	FIFO	RxUnderflow	0
		3	FIFO	RxOverflow	0
		4	CRC/Frame fail	CRCfail	1
		5	CRC/Frame fail	FrameFail	1
		6	CmdError	CmdError	0

The IRQmask registers are used to mask the different interrupt conditions. The reason for an interrupt request is indicated by the IRQreason register. If a bit is set in the mask register, the corresponding interrupt to the external MCU is disabled.

5.3 PHY Register

Note: The PHY registers should be changed only while in IDLE mode.

Register: PHY Current Channel [7:0]

mnemonic: RPCC, address 0x00, access R/W, C code-based constant address definition: PHY_CHANNEL

Addr.	Register	Channel	Frequency (MHz)	Reset
0x00	RPCC	0x00	868.3 (Europe)	0x00
		0x01	906 (North America)	
		0x02	908	
		0x03	910	
		0x04	912	
		0x05	914	
		0x06	916	
		0x07	918	
		0x08	920	
		0x09	922	
		0x0A	924, see note	
		0xFF	Reserved	

Reference: IEEE 802.15.4, Section 7.6.1.2

The RxMode register must be set according to the US or EU mode. Contact ZMD if applications desire non IEEE 802.15.4 compliant frequency channels. Check with local authorities for frequency regulations.

Note that setting the transceiver to channel 10 requires two additional register settings for best performance. After setting the channel to 10 by writing the value 0x0A to the RPCC register, then write 0xFF to the RPCC register and write the value 0x00 to register address 0x01.

Register: PHY Transmitter Mode Register [7:0]

mnemonic: RTXM, address 0x05, access R/W, C code-based constant address definition: TX_MODE

Addr.	Register	Bit	Description	Reset
0x05	RTXM	2:0	Tx mode: 0 = Normal 2 = Test Mode (continuous carrier) 4 = Test Mode (continuous frame with pn-code sequence)	0
		3	TXIO mode: 0 = RFIO 1 = RFO	0
		5:4	SEL_PA_LP (output power mode): 0 = 0 dBm 1 = -16 dBm 2 = -20 dBm 3 = -26 dBm	0
		7:6	Reserved	0

The Tx mode section is used to set the transmitter to normal operation mode or to continuous transmission mode for test purposes. For continuous Tx, the macTxConfig register must be set to DirectTx (0x04) and the transmission is initiated by the macControl command TxOn (0x03).

The TXIO mode bit determines whether the RFIO or the RFO pad is to be used as Tx output. The RFO pad is used with an external power amplifier. With the SEL_PA_LP, the transmitter output power can be reduced.

Register: PHY Receiver Mode Register [6:0]

mnemonic: RxMode, address 0x08, access R/W, C code-based constant address definition: RX_MODE

Addr.	Register	Bit	Description	Reset
0x08	RxMode	5:0	Reserved – DO NOT MODIFY	0x22
		6	RxUsMode	
		7	Not implemented	

Set RxMode[6] to 0 in EU and 1 in US mode. Do not modify bits [5:0].

Register: Energy Detection Threshold [7:0]

mnemonic: EdThreshold, address 0x24, access R/W, C code-based constant address definition: ED_THRESHOLD

Addr.	Register	Description	Reset
0x24	EdThreshold	Energy detection channel busy/idle threshold value	0x0F

Refer to section 4.7.

Register: AGC Level [7:0]
mnemonic: AgcLvl, address 0x0E, access R, C code-based constant address definition: AGC_LVL

Addr.	Register	Description	Reset
0x0E	AgcLvl	Automatic Gain Control Level	0x7F

The automatic gain control level is the variable part of the gain of the programmable gain amplifier (PGA). The measured AgcLvl value is above a certain dynamic range an indirect proportional input signal level. After a successfully received packet, it can be used to estimate the LQI (see section 4.6) and RSSI (see section 4.7). If the ContRx bit is set in the macRxConfig register, the AgcLvl of the last received frame cannot be read from this register. The AgcLvl is stored only until the receiver is switched on again.

5.4 MAC Operating Control

Register: MAC Control Commands [4:0]
mnemonic: macControl, address 0xA0, access R/W, C code-based constant address definition: MAC_CTRL
mnemonic: macControlT, address 0xA1, access R/W, C code-based constant address definition: MAC_CTRL_T

Addr.	Register	Command	Description	Reset
0xA0 0xA1	macControl macControlT	0x00 NoCmd *	macControl is ready for next command Set by hardware, indicating the acceptance and execution of the last command	(0x1F) 0x00
		0x01 Sleep	Start Sleep mode	
		0x02 TxRxOff	Turn off transceiver, go into Idle mode, power down analog front-end; refer to section 8	
		0x03 TxOn	Start transmitter	
		0x04 RxOn	Start receiver	
		0x05 ScanOn	Start scan mode	
		0x06 ScanOff	Stop scan (passive, active, orphan scan only)	
		0x07 ScanCont	Continue scan (passive, active, orphan scan only)	
		0x08 BcTrOn	Start beacon tracking (slave only)	
		0x09 BcTrSleep	Go into Sleep mode until next beacon (slave only)	
		0x0A BcTrOff	Stop beacon tracking	
		0x0B BcOk	Confirm the tracked beacon as valid (beacon tracking mode)	
		0x0C BcFail	Tracked beacon is not valid (beacon tracking mode)	
		0x0D AutoBcTxOn	Start automatic beacon generation (coordinator only)	
		0x0E AutoBcTxOff	Stop automatic beacon generation (coordinator only)	
		0x0F CsmaResetBcTx	Reset a paused CSMA transmission. Used for beacon transmitting devices.	
		0x10 CsmaResetBcTr	Reset a paused CSMA transmission. Used for beacon tracking devices.	
		0x11 PollResume	Resume the WaitForFrame Response process that was stopped by a data reception.	
		0x12 PollReset	Reset the WaitForFrame Response process	
		0x13 TxFifoFlush	Flush the TxFIFO	
		0x14 RxFifoFlush	Flush the RxFIFO	
		0x15 TotalTimerStart	Start the symbol time 24-bit global timer	
		0x16 TotalTimerStop	Stop the symbol time 24-bit global timer	
		0x17 GeneralTimerStart	Activate the T_General timer	
		0x18 GeneralTimerStop	Stop the T_General timer	
		0x19 TxFast	Request a transmission without CSMA that quickly follows an Ack frame. 802.15.4, 7.5.6.3	

Addr.	Register	Command	Description	Reset
		0x1F CmdError *	Command error; set by hardware, indicating an invalid command, either not defined or not appropriate in the current operating mode	

* Reserved by hardware.

The macControl register is the main control interface to the integrated HW-MAC. It uses a set of commands to control the HW-MAC operating modes. The NoCmd (0x00) and CmdError (0x1F) commands are reserved by the hardware for feedback information. If a command is written to the macControl register, hardware first checks whether it is valid in the current operating mode. If the command is not valid, the macControl is set to CmdError (0x1F). If the command is valid, it will remain in the macControl register until it can be accessed and executed. After it is accessed, the macControl is reset to NoCmd (0x00). The macControl register can queue one another command that is executed after the previous command has been completed.

For detailed information on the different operating modes and related commands refer to section *4.11 Operation Mode Description*.

Register: MAC Operating Mode [3:0]

mnemonic: macOpMode, address 0xA2, access R, C code-based constant address definition: MAC_OP_MODE

Addr.	Register	OpMode	Description	Reset
0xA2	macOpMode	0x00 Idle	Idle mode (analog front-end power down)	0x00
		0x01 PdnSleep	Sleep or Gobal Power Down mode (only 32.768 kHz clock running); this OpMode cannot be read via SPI	
		0x02 Tx	Transmit mode	
		0x03 RxIdle	Receiver Idle mode (analog Rx is on, digital Rx is idle)	
		0x04 RxActive	Receive mode (acquisition/reception running)	
		0x05 ScanTx	Scan mode Tx (active/orphan scan while transmitting)	
		0x06 ScanRx	Scan mode Rx (while scanning/receiving)	
		0x07 ScanAck	Scan mode Ack transmission (orphan scan realignment command acknowledgment)	
		0x08 BcTr	Beacon-Tracking mode (beacon scanning)	
		0x09 AutoBcTx	Automatic beacon transmission is ongoing	
		0x0A PhySwitch	Inter-operating mode step to switch the PHY state	

The macOpMode register monitors the current operating mode of the HW-MAC. Most of the operating mode changes are initiated by macControl commands. Note that the common sequence is previous operating mode -> PhySwitch -> new operating mode. The PhySwitch state lasts about 200 µs for Rx/Tx power up or turnaround. For details on the different operating modes, refer to chapter 4 Integrated HW-MAC.

Register: Power Down [7:0]
mnemonic: RPD, address 0x14, access R/W, C code-based constant address definition: R_PWR_DWN

Addr.	Register	Bit	Description	Reset
0x14	RPD	[6:0]	Reserved	0x3F
		7	Set this bit to switch off the 32 kHz oscillator with GPD	

The RPD register is used with the GPD signal to enter and leave the Off mode (refer to section 0 for further information on Off mode). It is required to keep the reserved bits unchanged while changing bit 7; i.e. write the value 0xBF to the register if the Off mode (32 kHz oscillator is switched off) is desired after setting GPD to high.

5.5 MAC FIFO Register

Register: Transmit FIFO [7:0]
mnemonic: TxFIFO, address 0x80, access W, C code-based constant address definition: TX_FIFO

Address	Register	Description	Reset
0x80	TxFIFO	First byte of the transmit FIFO	0xXX

Data written to this address is moved into the TxFIFO. Up to 128 bytes can be stored to the TxFIFO. The TxFIFO can contain more than one frame. After transmitting a frame, the corresponding data is removed from the TxFIFO. If the frame has not been sent due to a CSMA failure (busy channel), the frame is not removed from the TxFIFO and can be used for the next transmission or must be removed from the TxFIFO register using the macControl command TxFifoFlush.
The length of an IEEE 802.15.4 compliant frame is limited by the length field to 127 bytes. In DirectFifoAccess mode the integrated HW-MAC needs room for two bytes of the TxFIFO for CRC bytes that are appended at the end of the frame.

Register: Receive FIFO [7:0]
mnemonic: RxFIFO, address 0x81, access R/W, C code-based constant address definition: RX_FIFO

Address	Register	Description	Reset
0x81	RxFIFO	First byte of received data FIFO	0xXX

Data read from this address is shifted out from the RxFIFO. Data can also be written to the RxFIFO if the RxFIFO is used as the transmit FIFO for beacon transmission. To enable this mode, the macBcTxConfig[0] must be set. The RxFIFO register can be written only by a single byte access; it cannot be used with a multiple byte access, i.e., the length indicator of the SPI protocol can be one only by writing to the RxFIFO (see section 3.6 for further information on SPI protocol).

Register: MAC Tx/Rx FIFO Status [5:0]

mnemonic: macFifoStatus, address 0xA9, access R, C code-based constant address definition: MAC_FIFO_STATUS

Addr.	Register	Bit	Description	Reset
0xA9	macFifo Status	0	Tx Under/Overflow indicator	0
		1	TxEmpty	1
		2	TxFull	0
		3	Rx Under/Overflow indicator	0
		4	RxEmpty	1
		5	RxFull	0
		6	Not implemented	/
		7	Not implemented	/

The macFifostatus register monitors the state of the TxFIFO and RxFIFO. The TxFIFO is 128 bytes, and the RxFIFO is 256 bytes. The FIFO control logic blocks write access to a full FIFO and read access from an empty FIFO, and they also generate an interrupt in this case. For details, refer to section 4.3.

Register: Frame Pending Number [5:0]

mnemonic: macFramePend, address 0x88, access R, C code-based constant address definition: FIFO_FRM_PEND

Address	Register	Description	Reset
0x88	macFramePend	Number of pending frames in the MAC RxFIFO	0x00

This register contains the number of received frames queued in the RxFIFO. Each time a received frame is stored in the RxFIFO, the macFramePend number is increased. The register content is decreased by one every time the register is read.

5.6 MAC Tx Control

Register: MAC Transmitter Configuration [6:0]

mnemonic: macTxConfig, address 0xB1, access R/W, C code-based constant address definition: MAC_TX_CONFIG

Addr.	Register	Bit	Description	Reset
0xB1	macTxConfig	0	EnableSlottedMode	0
		[2:1]	TxOption: 00 = GTS 01 = CSMA (default) 10 = DirectTx	01
		3	GoToRx	0
		4	Enable Slotted Ack	1
		5	Battery Life Extension	0
		6	Direct Fifo Access	0
		7	Not implemented	/

This register configures the different transmit modes. It must be set up correctly before starting a transmission.

EnableSlottedMode
Enables the slotted mode. Slotted Tx can be done only within the active portion of the superframe. That requires that the automatic beacon transmission is active for a coordinator or that a slave has successfully tracked the beacon previously. All three modes - GTS, CSMA, and DirectTx - are possible in slotted mode. In unslotted mode, only CSMA or DirectTx are allowed.

TxOption
Configures whether the transmission is a CSMA a GTS or DirectTx transmission. A DirectTx transmission is immediately started when initiated by macControl.

GoToRx
If bit GoToRx is set, then the device will automatically do a Tx to Rx turnaround after the end of the transmission and then start the receiver. It is not necessary to set this bit if the acknowledge requested bit of the mhrFc1Tx register is set. It is necessary to set this bit if the ContRx bit and the AutoAckEnable bit of the macRxConfig register are set to continue receiving after an automatic acknowledgement transmission.

EnableSlottedAck
If this bit is set then the Ack transmission in slotted mode is aligned to a backoff period boundary, otherwise the slotted Ack is sent 12 symbols (default value, see T_Ack register) after the incoming frame as in the unslotted mode. This bit takes effect only if the EnableSlottedMode is enabled.

BatteryLifeExtension
This bit enables the CSMA battery life extension mode.

DirectFifoAccess
The DirectFifoAccess mode can be used to implement proprietary frame transmission. If this mode is enabled, some of the integrated HW-MAC features that are derived form the frame forming mechanism are disabled.
If this bit is set, then the integrated HW-MAC MAC header and address field construction is disabled. In this mode, the MAC header and address fields are taken directly from the TxFIFO. For further information on the

DirectFifoAccess mode see section 4.5. For details regarding the Tx operating mode, refer to the Tx operating mode description section 4.11.2.

Register: MAC Transmit Status [7:0]

mnemonic: macTxStatus, address 0xA3, access R, C code-based constant address definition: MAC_TX_STATUS

Addr.	Register	Status	Description	Reset
0xA3	macTxStatus	0x00 Off	Tx off	0x00
		0x01 GTS	GTS transmission	
		0x02 SltCSMA	Slotted CSMA transmission	
		0x03 SltDir	Slotted direct transmission	
		0x04 UnSltCSMA	Unslotted transmission	
		0x05 UnSltDir	Unslotted direct transmission	
		0x06 Ack	Automatic acknowledge transmission	
		0x07 Fast	Transmission quickly following an Ack	
		0x08 Success*	Transmission completed successfully	
		0x10 CAfail_Chbusy*	Transmission failed due to channel access failure because of a busy channel	
		0x20 CAfail_CAPfail*	Transmission failed due to channel access failure because of CAP end	
		0x40 GTSfail*	GTS transmission failed due to unavailable slot	
		0x80 CAfail_SFend*	Transmission failed due to super frame end	

* Status is indicated by the IRQ.

The macTxStatus register monitors the Tx status and the transmission result. Each result update is indicated by an interrupt. For details, refer to the Tx mode description section 4.11.2.

Register: Transmit Frame MSDU Length [6:0]

mnemonic: msduLengthTx, address 0x60, access R/W, C code-based constant address definition: MSDU_TX_LENGTH

Address	Register	Description	Reset
0x60	msduLengthTx	Transmit frame MAC payload length	0x00

Reference: IEEE 802.15.4, Section 5.4.3

The length of the MAC payload (MAC service data unit (MSDU)) must also be set to this register before a transmission is started by the macControl command TxOn. The MSDU itself is stored in the TxFIFO. This length value must only include the frame's payload and not the header length and not the CRC bytes length.
If the DirectFifoAccess mode is used, enabled by setting the DirectFifoAccess bit in the macTxConfig register, the length of the frame must be stored as the first byte into the TxFIFO and the msduLengthTx register is not used. See section 4.5 for further information on the DirectFifoAccess mode.
The register content is retained until a new value is assigned or a reset is applied.

Register: Transmit Frame MAC Header Frame Control Field [7:0]

mnemonic: mhrFc1Tx, address 0x61, access R/W, C code-based constant address definition: MHR_TX_FC_1

Addr.	Register	Bit	Description	Reset
0x61	mhrFc1Tx	0	Frame type (b0)	0
		1	Frame type (b1)	0
		2	Frame type (b2)	0
		3	Security enabled	0
		4	Frame pending	0
		5	Acknowledge requested	0
		6	IntraPan	0
		7	Reserved	0

Reference: IEEE 802.15.4, Section 7.2.1.1

The mhrFc1Tx register contains the lower byte (bit [7:0]) of the MAC header frame control field of the transmit frame. It must be configured before a transmission begins. For auto acknowledge generation, this register must be updated by the firmware before the acknowledge frame transmission begins.
The use of the DirectFifoAccess mode requires enabling the acknowledge request bit for the reception of an acknowledgement frame, if an automatic Tx/Rx switching should be performed. See section 4.5 for further information on the DirectFifoAccess mode.

Register: Transmit Frame MAC Header Frame Control Field [15:8]

mnemonic: mhrFc2Tx, address 0x62, access R/W, C code-based constant address definition: MHR_TX_FC_2

Addr.	Register	Bit	Description	Reset
0x62	mhrFc2Tx	0	Reserved	0
		1	Reserved	0
		2	Destination addressing mode (b0)	0
		3	Destination addressing mode (b1)	0
		4	Reserved	0
		5	Reserved	0
		6	Source addressing mode (b0)	0
		7	Source addressing mode (b1)	0

Reference: IEEE 802.15.4, Section 7.2.1.1

The mhrFc2Tx register contains the upper byte (bit [15:8]) of the MAC header frame control field of the transmit frame. It must be configured before a transmission begins. For auto acknowledge generation, this register is not used. The MAC header frame control field [15:8] is automatically set to 0 by the HW-MAC for acknowledge generation.

Register: Transmit Frame MAC Header Sequence Number Field [7:0]

mnemonic: mhrSquNbTx, address 0x63, access R/W, C code-based constant address definition: MHR_TX_SEQ_NO

Addr.	Register	Description	Reset
0x63	mhrSquNbTx	Transmit frame MAC header sequence number	0x00

Reference: IEEE 802.15.4, Section 7.2.1.2

The mhrSquNbTx register contains the sequence number of the transmit frame. It must be configured before a transmission begins. For auto acknowledge generation, the sequence number of the acknowledge frame is instead taken from the mhrSquNbRx register in order to use the sequence number of the previously received frame.

In the DirectFifoAccess mode the mhrSquNbTx register does not have an effect. The header of the frame must be written to the TxFIFO and the sequence number for the automatic sequence number checking is derived from the header bytes that are written to the TxFIFO.

Register: Tx Frame MAC Header Dest. PAN Identifier Field [15:0]

mnemonic: mhrDstPanIdTx, address 0x64 – 0x65, access R/W, C code-based constant address definition: MHR_TX_DST_PAN_ID_1 and MHR_TX_DST_PAN_ID_2

Addr.	Register	Description	Reset
0x64	mhrDstPanId1Tx	Transmit frame MAC header destination PAN identifier [7:0]	0x00
0x65	mhrDstPanId2Tx	Transmit frame MAC header destination PAN identifier [15:8]	0x00

Reference: IEEE 802.15.4, Section 7.2.1.3

This register stores the MAC header destination PAN identifier field for a transmit frame. The mhrFc2Tx addressing mode configuration determines whether the transmit frame address field contains the destination PAN identifier field. The addressing field is constructed for each frame within the HW-MAC frame former.

Register: Tx Frame MAC Header 16-bit Dest. Address Field [15:0]

mnemonic: mhrDstAddr16Tx, address 0x66 – 0x67, access R/W, C code-based constant address definition: MHR_TX_DST_ADDR16_1 and MHR_TX_DST_ADDR16_2

Addr.	Register	Description	Reset
0x66	mhrDstAddr16_1Tx	Transmit frame MAC header 16-bit destination address [7:0]	0x00
0x67	mhrDstAddr16_2Tx	Transmit frame MAC header 16-bit destination address [15:8]	0x00

Reference: IEEE 802.15.4, Section 7.2.1.4

This register stores the MAC header 16-bit destination address field for a transmit frame. The mhrFc2Tx addressing mode configuration determines whether the transmit frame address field contains the 16-bit destination address. The addressing field is constructed for each frame within the HW-MAC frame former.

Register: Tx Frame MAC Header 64-bit Dest. Address Field [63:0]

mnemonic: mhrDstAddr64Tx, address 0x68 – 0x6F, access R/W, C code-based constant address definition: MHR_TX_DST_ADDR64_1 through MHR_TX_DST_ADDR64_8

Addr.	Register	Description	Reset
0x68	mhrDstAddr64_1Tx	Transmit frame MAC header 64-bit destination address [7:0]	0x00
0x69	mhrDstAddr64_2Tx	Transmit frame MAC header 64-bit destination address [15:8]	0x00
0x6A	mhrDstAddr64_3Tx	Transmit frame MAC header 64-bit destination address [23:16]	0x00
0x6B	mhrDstAddr64_4Tx	Transmit frame MAC header 64-bit destination address [31:24]	0x00
0x6C	mhrDstAddr64_5Tx	Transmit frame MAC header 64-bit destination address [39:32]	0x00
0x6D	mhrDstAddr64_6Tx	Transmit frame MAC header 64-bit destination address [47:40]	0x00
0x6E	mhrDstAddr64_7Tx	Transmit frame MAC header 64-bit destination address [55:48]	0x00
0x6F	mhrDstAddr64_8Tx	Transmit frame MAC header 64-bit destination address [63:56]	0x00

Reference: IEEE 802.15.4, Section 7.2.1.4

This register stores the MAC header 64-bit destination address field for a transmit frame. The mhrFc2Tx addressing mode configuration determines whether the transmit frame address field contains the 64-bit destination address. The addressing field is constructed for each frame within the HW-MAC frame former.

Register: Transmit Frame MAC Header Source PAN Identifier Field [15:0]

mnemonic: mhrSrcPanIdTx, address 0x70 – 0x71, access R/W, C code-based constant address definition: MHR_TX_SRC_PAN_ID_1 and MHR_TX_SRC_PAN_ID_2

Addr.	Register	Description	Reset
0x70	mhrSrcPanId1Tx	Transmit frame MAC header source PAN identifier [7:0]	0x00
0x71	mhrSrcPanId2Tx	Transmit frame MAC header source PAN identifier [15:8]	0x00

Reference: IEEE 802.15.4, Section 7.2.1.5

This register stores the MAC header Source PAN identifier field for a transmit frame. The mhrFc2Tx addressing mode configuration determines whether the transmit frame address field contains the source PAN identifier field. The addressing field is constructed for each frame within the HW-MAC frame former.

Register: Transmit Frame MAC Header 16-bit Source Address Field [15:0]

mnemonic: mhrSrcAddr16Tx, address 0x72 – 0x73, access R/W, C code-based constant address definition: MHR_TX_SRC_ADDR16_1 and MHR_TX_SRC_ADDR16_2

Addr.	Register	Description	Reset
0x72	mhrSrcAddr16_1Tx	Transmit frame MAC header 16-bit source address [7:0]	0x00
0x73	mhrSrcAddr16_2Tx	Transmit frame MAC header 16-bit source address [15:8]	0x00

Reference: IEEE 802.15.4, Section 7.2.1.6

This register stores the MAC header 16-bit source address field for a transmit frame. The mhrFc2Tx addressing mode configuration determines whether the transmit frame address field contains the 16-bit source address. The addressing field is constructed for each frame within the HW-MAC frame former.

Register: Transmit Frame MAC Header 64-bit Source Address Field [63:0]

mnemonic: mhrSrcAddr64Tx, address 0x74 – 0x7B, access R/W, C code-based constant address definition: MHR_TX_SRC_ADDR64_1 through MHR_TX_SRC_ADDR64_8

Addr.	Register	Description	Reset
0x74	mhrSrcAddr64_1Tx	Transmit frame MAC header 64-bit source address [7:0]	0x00
0x75	mhrSrcAddr64_2Tx	Transmit frame MAC header 64-bit source address [15:8]	0x00
0x76	mhrSrcAddr64_3Tx	Transmit frame MAC header 64-bit source address [23:16]	0x00
0x77	mhrSrcAddr64_4Tx	Transmit frame MAC header 64-bit source address [31:24]	0x00
0x78	mhrSrcAddr64_5Tx	Transmit frame MAC header 64-bit source address [39:32]	0x00
0x79	mhrSrcAddr64_6Tx	Transmit frame MAC header 64-bit source address [47:40]	0x00
0x7A	mhrSrcAddr64_7Tx	Transmit frame MAC header 64-bit source address [55:48]	0x00
0x7B	mhrSrcAddr64_8Tx	Transmit frame MAC header 64-bit source address [63:56]	0x00

Reference: IEEE 802.15.4, Section 7.2.1.6

This register stores the MAC header 64-bit source address field for a transmit frame. The mhrFc2Tx addressing mode configuration determines whether the transmit frame address field contains the 64-bit source address. The addressing field is constructed for each frame within the HW-MAC frame former.

The 64-bit source address corresponds to the IEEE 64-bit extended unique identifier (EUI-64). The EUI-64 value is programmed into an on-chip EEPROM during production by ZMD. To update the mhrSrcAddr64Tx registers with the programmed EUI-64, the MTPcontrol register must be used. Refer to section 5.17 for further information on MTP handling.

Register: Transmit Frame CRC Field [15:0]

mnemonic: mfrCRCTx, address 0x7C – 0x7D, access R, C code-based constant address definition: MFR_TX_CRC_1 and MFR_TX_CRC_2

Address	Register	Description	Reset
0x7C	mfrCRC_1Tx	Transmitted frame CRC field [7:0]	0x00
0x7D	mfrCRC_2Tx	Transmitted frame CRC field [15:8]	0x00

Reference: IEEE 802.15.4, Section 7.2.1.8

These registers contain the CRC sequence of the last transmitted frame. They are for debugging and monitoring purposes.

Register: Transmit Timestamp [23:0]

mnemonic: macTxTime, address 0xF8 – 0xFA, access R, C code-based constant address definition: MAC_TX_TIME_1 though MAC_TX_TIME_3

Address	Register	Description	Reset
0xF8	macTxTime1	Timestamp of the transmitted frame [7:0]	0x00
0xF9	macTxTime2	Timestamp of the transmitted frame [15:8]	0x00
0xFA	macTxTime3	Timestamp of the transmitted frame [23:16]	0x00

The macTxTime registers contain the timestamp of the last transmitted frame. The timestamp is taken at the leading edge of the frame from the timer specified in the macTimerConfig register.

5.7 MAC Rx Control

Register: MAC Receiver Configuration [6:0]

mnemonic: macRxConfig, address 0xB2, access R/W, C code-based constant address definition: MAC_RX_CONFIG

Addr.	Register	Bit	Description	Reset
0xB2	macRxConfig	0	WaitResponseEnable	0
		1	AutoAckEnable	1
		2	ContRx	0
		3	FifoStoreAck	1
		4	FifoStoreLQI	1
		5	FifoStoreTimeStamp	0
		6	AckSquNbCheckEnable	1

WaitResponseEnable

This bit enables support for the MLME-POLL primitive (*Reference: IEEE 802.15.4, Section 7.1.16*). If the receiver is waiting for an acknowledgment and it receives an acknowledge frame, it will check the frame pending bit. If the frame pending bit is set and the WaitResponseEnable in macRxConfig register is set, it will expect the arrival of a data frame. In this case, it will continue receiving and wait for the arrival of the data frame for the time set in the T_MaxFrameResponse register.

AutoAckEnable

If the AutoAckEnable bit is set to 1, the integrated HW-MAC automatically transmits an acknowledge frame after the successful reception of a frame with the acknowledge requested bit set. Setting this bit in connection with the ContRx bit requires the setting of the GoToRx bit in the macTxConfig register for continuing unslotted receiving after an automatic acknowledgement transmitting.

ContRx

If the ContRx bit is set to 1, the receiver will continue receiving after the reception of a frame. Otherwise it will go to Idle mode after a frame reception. Do not set this bit for beacon tracking mode. Note that the AgcLvl is not stored in the AgcLvl register of the last received frame if this bit is set. Using the ContRx bit together with the AutoAckEnable bit requires the setting of the GoToRx bit in the macTxConfig register for continuing unslotted receiving mode after an automatic acknowledgement transmission.

FifoStoreAck

This bit enables the storage of acknowledge frames in the RxFIFO.

FifoStoreLQI

This bit enables the storage of a value to calculate the link quality indicator (LQI) after a frame in the RxFIFO. It is not recommended to use the value that is stored to the RxFIFO after the received frame to calculate the LQI value. Instead of that, the AgcLvl value should be used to calculate the LQI value. If the AgcLvl value is used to calculate the LQI value, the FifoStoreLQI bit can be set to 0.

FifoStoreTimeStamp
This bit enables the storage of a 3-byte receive timestamp after the frame in the RxFIFO.
If FifoStoreLQI is enabled, the timestamp is stored after the LQI.
An appropriate timer must be selected for the timestamp in the macTimerConfig register.

AckSquNbCheckEnable
If this bit is set, the sequence number of an incoming acknowledge frame is compared to the sequence number of the transmitted frame that requested the acknowledgment. Acknowledge frames for which the sequence number does not match are rejected.
In the DirectFifoAccess mode the sequence number for the automatic sequence number checking is derived from the header bytes that are written to the TxFIFO.

Register: MAC Receive Status [7:0]

mnemonic: macRxStatus, address 0xA4, access R, C code-based constant address definition: MAC_RX_STATUS

Addr.	Register	Status	Description	Reset
0xA4	macRxStatus	0x00 Off	Rx off	0x00
		0x01 Acquire	Acquisition is ongoing	
	(state)	0x02 RxActive	Frame reception is ongoing	
		0x10 Ack*	Acknowledge frame received	
		0x14 AckTimeOut*	Wait for acknowledge timed out	
		0x18 AckFramePend*	Ack received with frame pending bit set	
	(result)	0x20 PollNoData*	No frame received within T_MaxFrameResponse after an acknowledge with the frame pending bit set	
		0x24 PollCAPend*	No pending data received within the current CAP. WaitForFrame process will be summed in the next CAP.	
		0x80 Data*	Frame without the acknowledge requested bit set received	
		0xC0 DataAck*	Frame with the acknowledge requested bit set received	

* The result status is indicated by the IRQ.

The macRxStatus register is separated into two concurrent sections. The state section monitors the current status of the Rx state machine. The result section buffers the result of the last reception. The result section remains unchanged until the next result update. Each result update is indicated by an interrupt. For details, refer to the Rx mode description section 4.11.3.
If the macRxStatus register contains a value of 0x00 or 0x01 after a receive interrupt, it is necessary to check with the macFramePend register for a received frame in the RxFIFO.

Register: Received Frame MAC Header Frame Control Field [15:0]

mnemonic: mhrFcRx, address 0x82 – 0x83, access R, C code-based constant address definition: MHR_RX_FC_1 and MHR_RX_FC_2

Address	Register	Description	Reset
0x82	mhrFc1Rx	Received frame MAC header frame control field [7:0]	0x00
0x83	mhrFc2Rx	Received frame MAC header frame control field [15:8]	0x00

Reference: IEEE 802.15.4, Section 7.2.1.1

This register contains the MAC header frame control field of the last received frame and is debugging and monitoring purposes. Note that the received frames are queued in the RxFIFO (address 0x81), including MAC header and payload (refer to section 4.5).

Register: Received Frame MAC Header Sequence Number [7:0]

mnemonic: mhrSquNbRx, address 0x84, access R, C code-based constant address definition: MHR_RX_SEQ_NO

Address	Register	Description	Reset
0x84	mhrSquNbRx	Received frame MAC header sequence number	0x00

Reference: IEEE 802.15.4, Section 7.2.1.2

This register contains the MAC header sequence number of the last received frame. This sequence number is used in the transmitted acknowledge frames if auto acknowledge is enabled. Note that the received frames are queued in the RxFIFO (address 0x81), including MAC header and payload (refer to section 4.5).

Register: Received Frame MPDU Length [6:0]

mnemonic: mpduLengthRx, address 0x85, access R, C code-based constant address definition: MPDU_RX_LENGTH

Address	Register	Description	Reset
0x85	mpduLengthRx	Received frame mpdu length	0x00

Reference: IEEE 802.15.4, Section 5.4.3

This register contains the PHY payload (MAC protocol data unit (MPDU)) length of the last received frame including two CRC bytes.

Register: Received Frame CRC-Field [15:0]

mnemonic: mfrCRCRx, address 0x86 – 0x87, access R, C code-based constant address definition: MFR_RX_CRC_1 and MFR_RX_CRC_2

Address	Register	Description	Reset
0x86	mfrCRC1Rx	Received frame CRC field [7:0]	0x00
0x87	mfrCRC2Rx	Received frame CRC field [15:8]	0x00

Reference: IEEE 802.15.4, Section 7.2.1.8

These registers contain the CRC sequence of the last received frame.

Register: Receive Timestamp [23:0]
mnemonic: macRxTime, address 0xF5 – 0xF7, access R, C code-based constant address definition: MAC_RX_TIME_1 through MAC_RX_TIME_3

Address	Register	Description	Reset
0xF5	macRxTime1	Timestamp of the received frame [7:0]	0x00
0xF6	macRxTime2	Timestamp of the received frame [15:8]	0x00
0xF7	macRxTime3	Timestamp of the received frame [23:16]	0x00

The macRxTime contains the timestamp of the last successfully received frame. The timestamp is taken at the start of the MAC frame (end of SFD) of the received frame from the timer specified in the macTimerConfig register.

5.8 MAC Ack Control

Register: Transmit Ack Frame MAC Header Frame Control Field [4:0]
mnemonic: mhrAckFc1Tx, address 0x7E, access R/W, C code-based constant address definition: MHR_TX_ACK_FC_1

Addr.	Register	Bit	Description	Reset
0x7E	mhrAckFc1Tx	0	Security enabled	0
		1	Frame pending	0
		2	Acknowledge requested	0
		3	IntraPan	0
		4	Reserved	0

Reference: IEEE 802.15.4, Section 7.2.1.1
Note that the bit ordering within this register is not the same as it is when it appears later in the transmitted Ack frame. This register is used to control the acknowledgement frame.

Register: Transmit Ack Frame MAC Header Frame Control Field [7:0]

mnemonic: mhrAckFc2Tx, address 0x7F, access R/W, C code-based constant address definition: MHR_TX_ACK_FC_2

Addr.	Register	Bit	Description	Reset
0x7F	mhrAckFc2Tx	0	Reserved	0
		1	Reserved	0
		2	Destination addressing mode (b0)	0
		3	Destination addressing mode (b1)	0
		4	Reserved	0
		5	Reserved	0
		6	Source addressing mode (b0)	0
		7	Source addressing mode (b1)	0

For IEEE 802.15.4 compliance this register should not been changed.

Register: Frame to Acknowledge Space [5:0]

mnemonic: T_Ack, address 0xCC, access R/W, C code-based constant address definition: T_ACK

Address	Register	Description	Reset
0xCC	T_Ack	Space between a frame and an acknowledge in symbols	0x0C

Reference: IEEE 802.15.4, Section 7.5.1.2

This register sets the time in symbols between a frame requesting an acknowledgement and the returned acknowledge frame. The default value is 12. In order to be 802.15.4 compliant, it should not be changed. In slotted mode the Ack frame is deferred until the next backoff period boundary.

Register: Wait for Acknowledge Time [6:0]

mnemonic: T_WaitForAck, address 0xCF, access R/W, C code-based constant address definition: T_WAIT_FOR_ACK

Address	Register	Description	Reset
0xCF	T_WaitForAck	Wait time for an acknowledge in symbols	0x78

Reference: IEEE 802.15.4, Section 7.5

This register sets the maximum number in symbols to wait for the reception of an acknowledge frame after a transmitted data frame. It corresponds to macAckWaitDuration defined in the Standard. The default value is 120. In order to be 802.15.4 compliant, it should not be changed.

Register: Maximum Frame Response Wait Time [10:0]
mnemonic: T_MaxFrameResponse, address 0xD0 – 0xD1, access R/W, C code-based constant address definition: T_MAX_FRM_RESP_1 and T_MAX_FRM_RESP_2

Address	Register	Description	Reset
0xD0	T_MaxFrameResponse1	Maximum time to respond to a data request in symbols [7:0]	0xC4
0xD1	T_MaxFrameResponse2	Maximum time to respond to a data request in symbols [10:8]	0x04

Reference: IEEE 802.15.4, Section 7.5.6.3

These registers define the time in symbols that a device waits for a frame after the reception of an acknowledge frame with the FramePending bit set. This value corresponds to aMaxFrameResponseTime in the Standard. The default value is 1220 symbols. In order to be 802.15.4 compliant, it should not be changed.

5.9 MAC Scan Control

Register: MAC Scan Status [7:0]
mnemonic: macScanStatus, address 0xA5, access R, C code-based constant address definition: MAC_SCAN_STATUS

Addr.	Register	Status	Description	Reset
0xA5	macScanStatus	0x00 Off	Scan off	0x00
		0x01 ED	Energy detection scan active	
		0x02 ActiveTx	Active scan (request frame transmission ongoing)	
		0x03 ActiveRx	Active scan (scanning for beacon)	
		0x04 Passive	Passive scan	
		0x05 OrphanTx	Orphan scan (request frame transmission ongoing)	
		0x06 OrphanRx	Orphan scan (scanning for command frame)	
		0x08 Command*	Command frame received (orphan scan)	
		0x10 Beacon*	Beacon frame received (active/passive scan)	
		0x20 TimeOut*	T_ScanDuration timed out	
		0x40 ActiveTxFail*	Active scan request frame transmission failed due to channel access failure	
		0x80OrphanTxFail*	Orphan scan request frame transmission failed due to channel access failure	

* Status is indicated by the IRQ.

The macScanStatus register monitors the scan status. The successful reception of beacon or command frames during scan, the scan time out and the transmission failure of the request frames are indicated by interrupts. For details, refer to the scan mode description section 4.11.4.

Register: Maximum Measured ED-Scan Energy [7:0]
mnemonic: macScanED, address 0xA6, access R, C code-based constant address definition: MAC_SCAN_ED

Address	Register	Description	Reset
0xA6	macScanED	Maximum measured energy during an ED Scan	0x00

Reference: IEEE 802.15.4, Section 7.5.2.1.1

After an energy detection scan timeout, this register contains the maximum measured energy level of the last scan. The macScanED value can be assigned to an energy level. Refer to section 4.8 for further information on energy detection level.

Register: MAC Scan Mode Configuration [1:0]

mnemonic: macScanMode, address 0xB3, access R/W, C code-based constant address definition: MAC_SCAN_MODE

Addr.	Register	Bit	Description	Reset
0xB3	macScanMode	[1:0]	0 = energy detection scan 1 = active scan 2 = passive scan 3 = orphan scan	0
		[7:2]	Not implemented	/

Reference: IEEE 802.15.4, Section 7.5.2

This register configures the scan mode that is initiated with the macControl ScanOn command. For details, refer to the scan mode description in section 4.11.4.

Register: Scan Duration Time [23:0]

mnemonic: T_ScanDuration, address 0xD2 – 0xD4, access R/W, C code-based constant address definition: T_SCAN_DURATION_1 through T_SCAN_DURATION_3

Address	Register	Description	Reset
0xD2	T_ScanDuration1	Scan duration time [7:0] in symbols	0x00
0xD3	T_ScanDuration2	Scan duration time [15:8] in symbols	0x78
0xD4	T_ScanDuration3	Scan duration time [23:16] in symbols	0x00

Reference: IEEE 802.15.4, Section 7.5.2.1

These registers define the scan duration in symbols used for the different scan modes. For details, refer to the scan mode section 4.11.4. The default scan duration after reset is $960*2^5$ symbols. For debugging purposes, firmware can force a timeout of the running scan duration timer by setting T_ScanDuration to 0.

5.10 MAC Beacon Control

Register: Beacon Interval [23:0]

mnemonic: T_BeaconInterval, address 0xD5 – 0xD7, access R/W, C code-based constant address definition: T_BCN_INTERVAL_1 through T_BCN_INTERVAL_3

Address	Register	Description	Reset
0xD5	T_BeaconInterval1	Beacon interval time [7:0] in symbols	0x00
0xD6	T_BeaconInterval2	Beacon interval time [15:8] in symbols	0x78
0xD7	T_BeaconInterval3	Beacon interval time [23:16] in symbols	0x00

These registers define the beacon interval in symbols for the configuration of a beacon-enabled network. It should be set up correctly, according to the superframe configuration, before starting beacon generation or beacon tracking. For details, refer to the beacon generation and beacon tracking sections 4.11.6 and 4.11.5. The default beacon interval after reset is $960*2^5$ symbols; i.e., beacon order is five.

5.10.1 MAC Beacon Generation Control

Register: MAC Auto Beacon Transmit Configuration [0:0]

mnemonic: macBcTxConfig, address 0xB5, access R/W, C code-based constant address definition: MAC_BTX_CONFIG

Addr.	Register	Bit	Description	Reset
0xB5	macBcTxConfig	[0]	UseRxFIFO	0x00
		[7:1]	Reserved	

UseRxFIFO
If this bit is set, the RxFIFO is used for the beacon transmission instead of the TxFIFO. This mode is useful when the coordinator must prepare the next beacon transmission while another transmission is still pending in the TxFIFO. If the RxFIFO is used for beacon transmission, the integrated HW-MAC mac header and address field composition is disabled. The MAC header and address field is used directly from the RxFIFO.

Register: MAC Automatic Beacon Transmission Status [2:0]

mnemonic: macAutoBcTxStatus, address 0xA8, access R, C code-based constant address definition: MAC_AUTO_BTX_STATUS

Addr.	Register	Status	Description	Reset
0xA8	macAutoBcTxStatus	0x00 Off	Automatic beacon transmission is off	0x00
		0x01 Tx	Transmitting a beacon	
		0x02 Done	The beacon is transmitted and waiting for the next beacon.	
		0x04 TxReady*	Ready to transmit the next beacon Request firmware to prepare the next beacon	

* TxReady is indicated by the IRQ. Done IRQ is masked by default.

The macAutoBcTxStatus register monitors the current status of the automatic beacon Tx state machine. The auto beacon Tx cycle begins with the beacon transmission "Tx" and the status then changes to "Done". A time $2^{Td\_BeaconInterval}$ before the end of the beacon interval, an interrupt is generated and the status is updated to "TxReady," indicating that the HW-MAC is ready for the transmission of the next beacon and is requesting firmware to prepare the beacon. A new cycle begins with the start of the next beacon interval and the transmission of the next beacon. For details, refer to the automatic beacon Tx mode description in section 4.11.5.

Register: Beacon Interval Delta Time Exponent [3:0]

mnemonic: Td_BeaconInterval, address 0xD8, access R/W, C code-based constant address definition: T_BCN_INTERVAL_D

Addr.	Register	Description	Reset
0xD8	Td_BeaconInterval	Beacon interval delta time exponent [3:0]	0x04

This register is used for a coordinator in a beacon-enabled network. If the coordinator has enabled the auto beacon generation, an auto beacon Tx interrupt is generated at a time $2^{Td\_BeaconInterval}$ symbols before the transmission of the next beacon. The interrupt indicates to the firmware that it must prepare the next beacon frame. For details, refer to the beacon generation section 4.11.5. The default time after reset is 2^4 symbols. Firmware may need to adjust this value according to the microcontroller speed requirements.

Register: Beacon Transmit Timestamp [23:0]

mnemonic: macBeaconTxTime, address 0xF2 – 0xF4, access R, C code-based constant address definition: MAC_BCN_TX_TIME_1 through MAC_BCN_TX_TIME_3

Addr.	Register	Description	Reset
0xF2	macBeaconTxTime1	Timestamp of the transmitted beacon [7:0]	0x00
0xF3	macBeaconTxTime2	Timestamp of the transmitted beacon [15:8]	0x00
0xF4	macBeaconTxTime3	Timestamp of the transmitted beacon [23:16]	0x00

The macBeaconTxTime contains the timestamp of the last transmitted beacon. The timestamp is taken at the leading edge of the beacon frame from the timer specified in the macTimerConfig register.

5.10.2 MAC Beacon Tracking Control

Register: MAC Beacon Tracking Configuration [6:0]
mnemonic: macBcTrConfig, address 0xB4, access R/W, C code-based constant address definition: MAC_BTR_CONFIG

Addr.	Register	Bit	Description	Reset
0xB4	macBcTrConfig	0	TrackEnable	1
		1	ErrorLock	0
		2	EarlyGuard	0
		3	LateGuard	0
		4	BeaconConfirm	0
		5	CoordAddrCheck	1
		6	AlignEnable	1
		7	Not implemented	/

TrackEnable
This bit must be set to enable beacon tracking.

ErrorLock
After the 1st two consecutive beacons are tracked successfully, the beacon-tracking engine starts to correct the expected beacon interval by a timing error estimate. The timing error is caused by the crystal frequency deviation between coordinator and slave. The frequency deviation depends mostly upon the temperature and age of the crystal and can therefore be considered constant.
After a loss of synchronization, beacon tracking starts with a new beacon scan. If the ErrorLock bit is not set, it waits for the first two tracked beacons before making the first timing error estimate. If the timing error can be considered constant, the ErrorLock bit can be set in order to lock the last timing error estimate. Consequently, the first beacon interval after a recovered loss of synchronization is corrected with the locked timing error estimate.

EarlyGuard
This bit enables the early guard time window, which is used to compensate the timing error between coordinator and slave. The timing error becomes important for higher beacon orders and depends upon the crystal frequency deviations. During beacon tracking, the timing error is tracked and corrected. However, a residual error will remain for longer beacon intervals. The early guard time window delays the GTS transmission by the value defined in the T_Delta register.

LateGuard
This bit enables the late guard window. If enabled, the guard time T_Delta is added to the transaction length for CAP and GTS end checks. For details, refer to the beacon tracking mode and network timing sections 4.11.6 and 4.12 and 4.12.

BeaconConfirm
If this bit is set, the HW-MAC will wait for a confirmation of the tracked beacon by the firmware. The beacon is accepted by the firmware by sending the macControl command BcOk. The beacon is rejected by sending the macControl command BcFail.
By default the firmware beacon confirmation is disabled and the beacon is accepted automatically if it passes the integrated HW-MAC address check.

CoordAddrCheck
If this bit is set, then the source address of the incoming beacon is compared to the coordinator address stored in the macCoordSrcAddr1..2 or macCoordExtAddr1..8 register, in addition to the frame filter rules defined in IEEE 802.15.4, Section 7.5.6.2. If the source address of the beacon does not match, a PAN identifier conflict is likely and the beacon is not accepted. For details about the coordinator address check and PAN identifier conflict resolution refer to *IEEE 802.15.4, Section 7.5.2.2* and *7.5.4.1.*
A failing coordinator address check is indicated in the macBcTrStatus register. Note that beacon frames failing the coordinator address check are still stored in the RxFIFO for further processing by the firmware. If the CoordAddrCheck bit is not set, the source address of beacon frames is not compared to the coordinator address.

AlignEnable
This bit enables the integrated beacon interval timing correction. After each tracked beacon, the tracking error is derived and used to correct the next beacon interval duration. This timing correction reduces the timing drift between a coordinator and a device caused by different crystal frequency offsets.

Register: MAC Beacon Track Status [6:0]

mnemonic: macBcTrStatus, address 0xA7, access R, C code-based constant address definition: MAC_BTR_STATUS

Addr.	Register	Status	Description	Reset
0xA7	macBcTrStatus	0x00 Off	Beacon tracking is off	0x00
		0x01 Track	Tracking (waiting until the expected arrival of the next beacon, then starting scan for the beacon)	
		0x02 Scan	Scanning for the next beacon	
		0x04 SyncLoss*	Loss of synchronization (number macMaxLostBeacons of consecutive beacon missed)	
		0x08 Beacon*	Beacon successfully tracked	
		0x10 CorrdAddrFail	Beacon tracked with wrong SrcAddr (possible macPanID conflict)	
		0x20 Sync	Initial beacon tracked	
		0x40 Align	Consecutive beacon tracked (automatic timing correction is active)	

* Status indicated by the IRQ

The macBcTrStatus register monitors the beacon track status. The loss of synchronization and the arrival tracking of a beacon are indicated by interrupts. For details, refer to the beacon tracking mode description section 4.11.6.

Register: Coordinator Short Address [15:0]

mnemonic: macCoordSrcAddr, address 0x95 – 0x96, access R/W, C code-based constant address definition: COORD_SRC_ADDR_1 and COORD_SRC_ADDR_2

Addr.	Register	Description	Reset
0x95	macCoordSrcAddr1	Coordinator reference short address [7:0]	0x00
0x96	macCoordSrcAddr2	Coordinator reference short address [15:8]	0x00

Reference: IEEE 802.15.4, Section 7.5.4.1

If the CoordAddrCheck is enabled in the macBcTrConfig register, the short source address of the incoming beacon frames are compared to the macCoordSrcAddr to determine if the incoming beacon is meant for the device. See description of the macBcTrConfig register for further information.

Register: Coordinator Extended Address [63:0]

mnemonic: macCoordExtAddr, address 0x97 – 0x9E, access R/W, C code-based constant address definition: COORD_EXT_ADDR_1 through COORD_EXT_ADDR_8

Addr.	Register	Description	Reset
0x97	macCoordExtAddr1	Coordinator reference extended address [7:0]	0x00
0x98	macCoordExtAddr2	Coordinator reference extended address [15:8]	0x00
0x99	macCoordExtAddr3	Coordinator reference extended address [23:16]	0x00
0x9A	macCoordExtAddr4	Coordinator reference extended address [31:24]	0x00
0x9B	macCoordExtAddr5	Coordinator reference extended address [39:32]	0x00
0x9C	macCoordExtAddr6	Coordinator reference extended address [47:40]	0x00
0x9D	macCoordExtAddr7	Coordinator reference extended address [55:48]	0x00
0x9E	macCoordExtAddr8	Coordinator reference extended address [63:56]	0x00

Reference: IEEE 802.15.4, Section 7.5.4.1

If the CoordAddrCheck is enabled in the macBcTrConfig register, the extended source address of the incoming beacon frames are compared to the macCoordExtAddr to determine if the incoming beacon is meant for the device. See description of the macBcTrConfig register for further information.

Register: Maximum Lost Beacon Number [3:0]

mnemonic: macMaxLostBeacons, address 0xC3, access R/W, C code-based constant address definition: BTR_MAX_LOST_BCNS

Addr.	Register	Description	Reset
0xC3	macMaxLostBeacons	Maximum number of lost beacons before a loss of synchronization indication	0x04

Reference: IEEE 802.15.4, Section 7.5

This register defines the number of consecutive beacons that will cause the internal HW-MAC beacon-tracking algorithm to generate an IRQ indicating the loss of synchronization. Refer to the beacon tracking section 4.11.6. The default value is 4. In order to be 802.15.4 compliant, it should not be changed.

Register: Current Number of Lost Beacon Number [3:0]

mnemonic: macSyncLoss, address 0xC4, access R, C code-based constant address definition: BTR_SYNC_LOSS

Addr.	Register	Description	Reset
0xC4	macSyncLoss	Current number of lost beacons	0x04

Reference: IEEE 802.15.4, Section 7.5

This register monitors the current number of lost beacons during beacon tracking. If this number reaches macMaxLostBeacons, the beacon-tracking engine generates a SyncLoss interrupt. The macSyncLoss is reset to 0 after each successfully tracked beacon. For details, refer to the beacon tracking section 4.11.6.

Register: Beacon Tracking Timing Correction Threshold [3:0]

mnemonic: BcTrThreshold, address 0xC7, access R/W, C code-based constant address definition: BTR_THRESHOLD

Addr.	Register	Description	Reset
0xC7	BcTrThreshold	Beacon tracking error threshold to enable the beacon interval timing correction.	0x01

If the difference between the estimated beacon interval in the tracking device and the measured beacon interval of the coordinator is greater than this threshold, the estimated beacon interval is corrected by the measured difference.

Register: Beacon Scan Duration Time [23:0]

mnemonic: T_BeaconScanDuration, address 0xD9 – 0xDB, access R/W, C code-based constant address definition: T_BCN_SCAN_DUR_1 through T_BCN_SCAN_DUR_3

Addr.	Register	Description	Reset
0xD9	T_BeaconScanDuration1	Beacon scan duration time [7:0] in symbols	0x00
0xDA	T_BeaconScanDuration2	Beacon scan duration time [15:8] in symbols	0xF0
0xDB	T_BeaconScanDuration3	Beacon scan duration time [23:16] in symbols	0x00

These registers define the beacon scan duration in symbols used in the beacon-tracking mode. For details, refer to the beacon-tracking mode section 4.11.6. The default beacon scan duration after reset is $960*2^6$ symbols (i.e. about 3 seconds for EU mode). For debugging purposes, firmware can force a timeout of the running beacon scan duration timer by setting T_BeaconScanDuration to 0.

Note that for active and passive scans, the scan duration time is set in the T_ScanDuration register.

Register: Beacon Scan Start Time [10:0]

mnemonic: T_BeaconScanStart, address 0xDC – 0xDD, access R/W, C code-based constant address definition: T_BCN_SCAN_START_1 and T_BCN_SCAN_START_2

Addr.	Register	Description	Reset
0xDC	T_BeaconScanStart1	Beacon scan start time [7:0] in symbols	0x0A
0xDD	T_BeaconScanStart2	Beacon scan start time [10:8] in symbols	0x00

These registers are used in beacon-tracking mode. In beacon-tracking mode, the HW-MAC starts to scan for the next beacon T_BeaconScanStart symbols before the expected arrival of the beacon. The default value after reset is 10 symbols. For details, refer to the beacon tracking mode section 4.11.6.

Register: Timing Error Guard Time [10:0]

mnemonic: T_Delta, address 0xE1 – 0xE2, access R/W, C code-based constant address definition: T_DELTA_1 and T_DELTA_2

Addr.	Register	Description	Reset
0xE1	T_Delta1	Timing error guard time [7:0] in symbols	0x00
0xE2	T_Delta2	Timing error guard time [10:8] in symbols	0x00

The T_Delta time defines the size of the guard time window to compensate for the timing error between the coordinator and the slave introduced by the crystal frequency deviation. The timing error becomes important for higher beacon orders. A beacon order 7 and a frequency deviation of 40 ppm results in $960 * 2^7 * 40ppm = 5$ symbols timing error at the end of a beacon interval. The guard time window must be enabled by setting the EarlyGuard and LateGuard bits in the macBcTrConfig register to (1). The guard time window can be activated separately for early and late checks. For early guard, the start of GTS transmission is delayed by the T_Delta time. For late guard, the T_Delta time is added to the calculated transaction time during CAP/GTS end check. The guard time should be activated only in beacon-enabled networks for slave devices.

Register: Last Received Beacon Tracking Error [12:0]

mnemonic: macBeaconTrackError, address 0xED – 0xEE, access R, C code-based constant address definition: MAC_BCN_TRACK_ERR_1 and MAC_BCN_TRACK_ERR_2

Addr.	Register	Description	Reset
0xED	macBeaconTrackError1	Arrival time of the last tracked beacon [7:0] in RTC units	0x00
0xEE	macBeaconTrackError2	Arrival time of the last tracked beacon [12:8] in RTC units	0x00

* One RTC unit is 2/(32.768 kHz) in EU and 1/(32.768 kHz) US mode.

The macBeaconTrackError contains the arrival time of the last tracked beacon relative to the expected arrival time. It is updated at the end of each successfully tracked beacon and is an indicator of the current beacon tracking timing error. The value is signed in 2's complement representation. Positive means an early beacon arrival and negative means a late arrival. Note that the unit is not in symbols but in 1/(32.768 kHz) units for US and 2/(32.768 kHz) units for EU. For more details on beacon tracking, refer to section 4.11.6.

Register: Beacon Receive Timestamp [23:0]
mnemonic: macBeaconRxTime, address 0xEF – 0xF1, access R, C code-based constant address definition: MAC_BCN_RX_TIME_1 through MAC_BCN_RX_TIME_3

Addr.	Register	Description	Reset
0xEF	macBeaconRxTime1	Timestamp of the received beacon [7:0]	0x00
0xF0	macBeaconRxTime2	Timestamp of the received beacon [15:8]	0x00
0xF1	macBeaconRxTime3	Timestamp of the received beacon [23:16]	0x00

The macBeaconRxTime contains the timestamp of the last tracked beacon. The timestamp is taken at the leading edge of the beacon frame from the timer specified in the macTimerConfig register.

5.11 MAC Timer Control and Values

Register: MAC Timer and Control Status [4:0]
mnemonic: macTimerControlStatus, address 0xAB, access R, C code-based constant address definition: MAC_TMR_CTRL_STATUS

Addr.	Register	Bit	Description	Reset
0xAB	macTimerControlStatus	0	WakeUp*	0x00
		1	GeneralExp*	
		2	macControlExp*	
		3	CmdErrorT*	
		4	CmdError*	

*Status is indicated by the IRQ.

Register: MAC General Purpose Timer Configuration [3:0]
mnemonic: macTimerConfig, address 0xB7, access R/W, C code-based constant address definition: MAC_TIMER_CONFIG

Addr.	Register	Bit	Description	Reset
0xB7	macTimerConfig	[1:0]	TimerSelection:	0x00
			0 – T_TotalTimeFFD	
			1 – T_Superframe	
			2 –T_TotalTimeRFD	
		[3:2]	TimestampSelection	
			0 – T_TotalTimeFFD	
			1 – T_Superframe	
			2 –T_TotalTimeRFD	

TimerSelection
Selects the timer that is used for the general purpose timer T_General interrupt or for the timer-controlled macControlT command execution.

TimestampSelection
Selects the timer from which the receive and transmit timestamp is derived.

Timer
T_TotalTimeFFD

This is a 24-bit timer with symbol time resolution (EU mode: 50 µs; US mode: 25 µs) and is driven by the external 24 MHz crystal. It either is started automatically by the beacon transmission or it needs to be started/stopped by the macControl command TotalTimerStart/TotalTimerStop in nonbeacon mode operation.

Note that during Sleep and Global Power Down modes, the 24 MHz crystal oscillator is stopped, and as a result this timer is paused.

T_TotalTimeRFD

This timer is driven by the 32.768 kHz crystal and has a resolution of 1/(32.768 kHz) (US) or 2/(32.768 kHz) (EU). The timer is available only in beacon tracking mode. It is a count-down timer, which counts down from the estimated beacon interval to zero.

T_Superframe

This timer counts the symbols of the active portion of the superframe starting from zero. It is available only in slotted mode.

Register: macControlT Command Execution Time [23:0]

mnemonic: T_MacControl, address 0xC9 – 0xCB, access R/W, C code-based constant address definition: T_MAC_CTRL_1 through T_MAC_CTRL_3

Addr.	Register	Description	Reset
0xC9	T_MacControl1	macControlT execution time [7:0]	0x00
0xCA	T_MacControl2	macControlT execution time [15:8]	0x00
0xCB	T_MacControl3	macControlT execution time [23:16]	0x00

These registers set the execution time of the MAC control command buffered in macControlT.
The timer from which the execution time is derived must be running and can be selected from the macTimerConfig register. For details, refer to the general purpose timer section 4.13.3.

Register: General Purpose Timer and Sleep Time [23:0]

mnemonic: T_General, address 0xDE – 0xE0, access R/W, C code-based constant address definition: T_GENERAL_1 through T_GENERAL_3

Addr.	Register	Description	Reset
0xDE	T_General1	Sleep time [7:0] in RTC units or general purpose timer	0x00
0xDF	T_General2	Sleep time [15:8] in RTC units or general purpose timer	0x00
0xE0	T_General3	Sleep time [23:16] in RTC units or general purpose timer	0x00

* One RTC unit is 2/(32.768 kHz) in EU and 1/(32.768 kHz) US mode.

These register define the sleep time duration in RTC units or the timer value of the general purpose timer T_General. The source of the general purpose timer is selected in the macTimerConfig register. For details, refer to the Sleep mode section 0.

Register: Current FFD Network Time [23:0]

mnemonic: macTotalTimeFFD, address 0xE3 – 0xE5, access R, C code-based constant address definition: MAC_TOT_TIME_FFD_1 through MAC_TOT_TIME_FFD_3

Addr.	Register	Description	Reset
0xE3	macTotalTimeFFD1	Current network time of a FFD [7:0] in symbols	0x00
0xE4	macTotalTimeFFD2	Current network time of a FFD [15:8] in symbols	0x00
0xE5	macTotalTimeFFD3	Current network time of a FFD [23:16] in symbols	0x00

If the device is a FFD and automatic beacon generation is active, then these registers will monitor the current network time between two beacons. The beginning of the beacon interval is assigned to the value T_BeaconInterval – 1 and the end to 0. The unit is in symbols.
If the device does not use beacon generation (i.e., unslotted mode), this timer must be initiated by the macControl command TotalTimerStart.

Register: Current RFD Network Time [23:0]

mnemonic: macTotalTimeRFD, address 0xE6 – 0xE8, access R, C code-based constant address definition: MAC_TOT_TIME_RFD_1 through MAC_TOT_TIME_RFD_3

Addr.	Register	Description	Reset
0xE6	macTotalTimeRFD1	Current network time of a RFD [7:0] in RTC units	0x00
0xE7	macTotalTimeRFD2	Current network time of a RFD [15:8] in RTC units	0x00
0xE8	macTotalTimeRFD3	Current network time of a RFD [23:16] in RTC units	0x00

* One RTC unit is 2/(32.768 kHz) in EU and 1/(32.768 kHz) US mode.

If the device is an RFD and a beacon tracking has been activated, then these registers will monitor the current network time between two beacons. The timer is restarted after each tracked beacon. The initial value is the current network time at the end of the tracked beacon minus an error estimate derived by the HW-MAC from the previous beacon interval. The error estimate is used to correct the timing deviation caused by frequency offset between the crystals of the coordinator and the slave. The expected end of the current beacon interval is when the macTotalTimeRFD registers are 0. For more details on beacon tracking, refer to section 4.11.6.

The RFD uses the RTC clock to maintain its network time, allowing the device to go into Sleep mode and Global Power Down mode the 24 MHz crystal between two beacons without loosing track of the network time. Therefore, macTotalTimeRFD is derived from the RTC.

Note that its unit is 2/(32.768 kHz) in EU and 1/(32.768 kHz) in US.

5.12 MAC Frame Filter Control

Register: MAC Frame Filter Configuration [3:0]

mnemonic: macFilterConfig, address 0xB6, access R/W, C code-based constant address definition: MAC_FILT_CONFIG

Addr.	Register	Bit	Description	Reset
0xB6	macFilterConfig	0	Lvl1FilterEnable	1
		1	Lvl2FilterEnable	1
		2	Lvl3FilterEnable	1
		3	PanCoordinator	0
		4	ReservedFrameTypeEnable	0
		5	ReservedFrameTypeFilterEnable	0

Lvl1FilterEnable
Enable CRC filter. Frames with CRC failure are ignored and removed from the RxFIFO. Use of the default filter configuration is recommended in order to be IEEE 802.15.4 compliant. For proprietary solutions, it is recommended that at least this bit is set to avoid false alarms.

Lvl2FilterEnable
Enables frame type and address filter. Frames with illegal frame type and illegal addresses are ignored and removed from the RxFIFO. The filter rules in accordance with IEEE 802.15.4, Section 7.5.6.2.

Lvl3FilterEnable
Enable operating mode depending upon frame type filtering (e.g., scan, beacon track and waiting for acknowledge).

PanCoordinator
Indicates to the address filter that the device is operating as a PAN coordinator.

ReservedFrameTypeEnable
Enables the reception of frames with reserved frame types. This is for upward compatibility.

ReservedFrameTypeFilterEnable
If this bit is set, the filter rules, in accordance with IEEE 802.15.4, Section 7.5.6.2, are applied for frames with a reserved frame type.

Register: MAC Frame Filter Status [2:0]

mnemonic: macFilterStatus, address 0xAA, access R, C code-based constant address definition: MAC_FILT_STATUS

Addr.	Register	Bit	Description	Reset
0xAA	macFilterStatus	0	CRC failure*	0x00
		1	Address/Frame Type failure*	
		2	Coordinator address failure	

*Can be indicated by the IRQ. Frame failure and CRC failure interrupts are masked by default.

Register: Reference MAC PAN Identifier [15:0]

mnemonic: macPanId, address 0x89 – 0x8A, access R/W, C code-based constant address definition: AF_PAN_ID_1 and AF_PAN_ID_2

Addr.	Register	Description	Reset
0x89	macPanId1	Frame filter reference MAC PAN identifier [7:0]	0x00
0x8A	macPanId2	Frame filter reference MAC PAN identifier [15:8]	0x00

Reference: IEEE 802.15.4, Section 7.5.6.2

The macPanId is used as reference MAC PAN identifier by the frame filter.

Register: Reference MAC Short Address [15:0]

mnemonic: macShortAddr, address 0x8B – 0x8C, access R/W, C code-based constant address definition: AF_ADDR16_1 and AF_ADDR16_2

Addr.	Register	Description	Reset
0x8B	macShortAddr1	Frame filter reference MAC short address [7:0]	0x00
0x8C	macShortAddr2	Frame filter reference MAC short address [15:8]	0x00

Reference: IEEE 802.15.4, Section 7.5.6.2

The macShortAddr is used as reference MAC short address by the frame filter.

Register: aExtendedAddr – Reference Extended Address [63:0]

mnemonic: aExtendedAddr, address 0x8D – 0x94, access R/W, C code-based constant address definition: AF_ADDR64_1 through AF_ADDR64_8

Addr.	Register	Description	Reset
0x8D	aExtendedAddr1	Frame filter reference extended address [7:0]	0x00
0x8E	aExtendedAddr2	Frame filter reference extended address [15:8]	0x00
0x8F	aExtendedAddr3	Frame filter reference extended address [23:16]	0x00
0x90	aExtendedAddr4	Frame filter reference extended address [31:24]	0x00
0x91	aExtendedAddr5	Frame filter reference extended address [39:32]	0x00
0x92	aExtendedAddr6	Frame filter reference extended address [47:40]	0x00
0x93	aExtendedAddr7	Frame filter reference extended address [55:48]	0x00
0x94	aExtendedAddr8	Frame filter reference extended address [63:56]	0x00

Reference: IEEE 802.15.4, Section 7.5.6.2

The aExtendedAddr is used as reference extended address by the frame filter.

If the ZWIR4501's unique IEEE address is required to use for the frame filter as well, the values of the mhrSrcAddr64Tx registers must be written to the aExtendedAddrX registers.

5.13 MAC Superframe and GTS Control

Register: Superframe Order (SO) [3:0]

mnemonic: macSuperframeOrder, address 0xBF, access R/W, C code-based constant address definition: SF_ORDER

Addr.	Register	Description	Range	Reset
0xBF	macSuperframeOrder	Superframe order, active portion of the superframe	0-14	0x05

Reference: IEEE 802.15.4, Section 7.5.1.1

This register sets the active portion of the superframe in a beacon-enabled network. The active portion is the SuperframeDuration (SD):

SD = 960 * 2^{SO} symbols

SD must be smaller than or equal to the beacon interval T_BeaconInterval.

Note that the value 15 is not allowed. In 802.15.4, the value 15 is used to disable the superframe. In the ZWIR4501, the enable slotted bit must be set to zero in the macTxConfig register to disable any slotted transmission.

Register: Contention Access Period (CAP) End [3:0]

mnemonic: macCAPend, address 0xC0, access R/W, C code-based constant address definition: SF_CAP_END

Addr.	Register	Description	Range	Reset
0xC0	macCAPend	Last slot of the contention access period	0-15	0x0F

Reference: IEEE 802.15.4, Section 7.5.1.1

This register defines the last slot of the CAP within the superframe.

Register: Guaranteed Time Slot (GTS) Start [3:0]

mnemonic: macGTSstart, address 0xC1, access R/W, C code-based constant address definition: SF_GTS_START

Addr.	Register	Description	Range	Reset
0xC1	macGTSstart	First slot of the guaranteed time slot period	0-15	0x0A

Reference: IEEE 802.15.4, Section 7.5.1.1

This register defines the first slot of the GTS within the superframe. If macGTSlength is set to 0, then no GTS is assigned and the macGTSstart value is ignored. If macGTSlength is greater than 0, then macGTSstart must be greater than macCAPend.

Register: Guaranteed Time Slot (GTS) Length [3:0]

mnemonic: macGTSlength, address 0xC2, access R/W, C code-based constant address definition: SF_GTS_LENGTH

Addr.	Register	Description	Range	Reset
0xC2	macGTSlength	Length of the GTS in slots	0-15	0x00

Reference: IEEE 802.15.4, Section 7.5.1.1

This register defines the length in slots of the GTS within the superframe. If macGTSlength is set to 0, then no GTS is assigned. This value must not exceed the number of slots from the beginning of the GTS to slot 15.

Register: Current Symbol Time [23:0]

mnemonic: macCurrentSymbolTime, address 0xE9 – 0xEB, access R, C code-based constant address definition: MAC_CURR_SYM_TIME_1 through MAC_CURR_SYM_TIME_3

Addr.	Register	Description	Reset
0xE9	macCurrentSymbolTime1	Current symbol time within the active superframe portion [7:0] in symbols	0x00
0xEA	macCurrentSymbolTime2	Current symbol time within the active superframe portion [15:8] in symbols	0x00
0xEB	macCurrentSymbolTime3	Current symbol time within the active superframe portion [23:16] in symbols	0x00

The macCurrentSymbolTime monitors the value of the internal superframe timer and shows the current point in time within the active portion of the superframe. In the case of an FFD, this timer starts with 0 at the beginning of each transmitted beacon. In the case of an RFD, the timer starts at the end of a tracked beacon and the start value is equal to the length of the beacon frame. It counts to the end of slot number 15.

Register: Current Slot Number [3:0]

mnemonic: macCurrentSlot, address 0xEC, access R, C code-based constant address definition: MAC_CURR_SLOT

Addr.	Register	Description	Range	Reset
0xEC	macCurrentSlot	Current slot within the active superframe portion	0-15	0x00

This register monitors the current slot number within the active portion of the superframe. It is derived from the internal superframe timer and requires an FFD generating beacon or an RFD successfully tracking beacon.

Register: Maximum Short Interframe Space (SIFS) Frame Size [6:0]

mnemonic: macMaxSIFSFrameSize, address 0xC5, access R/W, C code-based constant address definition: BTR_MAX_SIFS_FRM_SIZE

Addr.	Register	Description	Reset
0xC5	macMaxSIFSFrameSize	Maximum MPDU size in octets that can be followed by an SIFS	0x12

Reference: IEEE 802.15.4, Section 7.5

The macMaxSIFSFrameSize sets the maximum MPDU size in octets that can be followed by an SIFS period. The default value is 18. In order to be 802.15.4 compliant, it should not be changed.

Register: Superframe Alignment Order [3:0]

mnemonic: SFalignOrder, address 0xC6, access R/W, C code-based constant address definition: SF_ALIGN_ORDER

Addr.	Register	Description	Reset
0xC6	SFalignOrder	Superframe alignment order [3:0]	0x0C

The HW-MAC uses the 32.768 kHz RTC to maintain the network time between two beacons. This enables a slave device to go into Sleep mode and to Global Power Down mode the 24 MHz crystal at any time within a superframe without loosing the network time. The timing error between the coordinator and the slave introduced by the crystal frequency deviation is tracked and corrected by the internal HW-MAC RTC unit. For a slave device, the time within the active portion of the superframe, as well as the slots, is derived from a superframe timer which is fed by the 24 MHz crystal. For higher beacon order, the superframe timer must be aligned periodically with the RTC timer in order to limit the timing error. This alignment is done automatically within a superframe every $30 * 2^{SFalignOrder}$ symbols. Depending upon the frequency deviation, the SFalignOrder should be set to a value so that the first alignment happens when the timing error becomes more than 3 symbols. For details, refer to section 4.12.3.

Register: Short Inter-Frame Spacing [5:0]

mnemonic: T_SIFS, address 0xCD, access R/W, C code-based constant address definition: T_SIFS

Addr.	Register	Description	Reset
0xCD	T_SIFS	Short inter-frame spacing in symbols	0x0C

Reference: IEEE 802.15.4, Section 7.5.1.2

This register sets the length in symbols of the inter-frame spacing that follows frames with an MPDU size up to macMaxSIFSFrameSize. If the frame is followed by an acknowledgment, the inter-frame spacing comes after the acknowledge frame. The default value is 12. For 802.15.4 compliance, this value must be at least 12.
Note that, in DirectFifoAccess (macTxConfig) mode the IFS following the transmit frame should be calculated by SW and both the T_SIFS and the T_LIFS register should be temporarily overwritten by the calculated IFS.

Register: Long Inter-Frame Spacing [5:0]

mnemonic: T_LIFS, address 0xCE, access R/W, C code-based constant address definition: T_LIFS

Addr.	Register	Description	Reset
0xCE	T_LIFS	Long inter-frame spacing in symbols	0x28

Reference: IEEE 802.15.4, Section 7.5.1.2

This register sets the length in symbols of the inter-frame spacing that follows frames with an MPDU greater than macMaxSIFSFrameSize octets. If the frame is followed by an acknowledgment, the inter-frame spacing comes after the acknowledge frame. The default value is 40. For 802.15.4 compliance, this value must be at least 40.
Note that, in DirectFifoAccess (macTxConfig) mode the IFS following the transmit frame should be calculated by SW and both the T_SIFS and the T_LIFS register should be temporarily overwritten by the calculated IFS.

5.14 MAC CSMA Control

Register: Unit Backoff Period [7:0]

mnemonic: macUnitBackOffPeriod, address 0xC8, access R/W, C code-based constant address definition: BTR_UNIT_BACKOFF_PRD

Addr.	Register	Description	Reset
0xC8	macUnitBackOffPeriod	Number of symbols comprising the basic CSMA-CA backoff period	0x14

Reference: IEEE 802.15.4, Section 7.5

The macUnitBackOffPeriod defines the length of a backoff period used by the CSMA-CA algorithm. The default value is 20. In order to be 802.15.4 compliant, it should not be changed.

Register: Minimum CSMA-CA Backoff Exponent [2:0]

mnemonic: macMinBe, address 0xB9, access R/W, C code-based constant address definition: CSMA_MIN_BE

Addr.	Register	Description	Reset
0xB9	macMinBe	Minimum backoff exponent used by CSMA-CA	0x02

Reference: IEEE 802.15.4, Section 7.5

The macMinBE sets the minimum backoff exponent used by the CSMA-CA algorithm. In order to run the CSMA-CA algorithm in Battery Life Extension mode, macMinBE must be set to a value smaller than 3. Note that this register is initialized to 2; however, the default value in the standard is 3. Software might have to initialize this value during start-up. Note that the range defined by the 802.15.4 standard is 0 to 3; however, the HW-MAC allows values up to 7.

Register: Maximum CSMA-CA Backoff Exponent [2:0]

mnemonic: macMaxBe, address 0xBA, access R/W, C code-based constant address definition: CSMA_MAX_BE

Addr.	Register	Description	Reset
0xBA	macMaxBe	Maximum backoff exponent used by CSMA-CA	0x05

Reference: IEEE 802.15.4, Section 7.5

The macMaxBE sets the upper boundary for the backoff exponent used by the CSMA-CA algorithm. The default value is 5. In order to be 802.15.4 compliant, it should not be changed.

Register: CSMA_CA Initial Contention Window Size [2:0]

mnemonic: macInitialCW, address 0xBB, access R/W, C code-based constant address definition: CSMA_INITIAL_CW

Addr.	Register	Description	Reset
0xBB	macInitialCW	Initial size of the CSMA-CA contention window	0x02

Reference: IEEE 802.15.4, Section 7.5

The macInitialCW sets the initial contention window size used by the CSMA-CA algorithm. The default value is 2. In order to be 802.15.4 compliant, it should not be changed.

Register: Maximum Number of CSMA-CA Backoffs [2:0]

mnemonic: macMaxCSMABackOffs, address 0xBC, access R/W, C code-based constant address definition: CSMA_MAX_CSMA_BACKOFFS

Addr.	Register	Description	Reset
0xBC	macMaxCSMABackOffs	Maximum number of backoffs done by CSMA-CA	0x04

Reference: IEEE 802.15.4, Section 7.5

The macMaxCSMABackOffs sets the maximum number of backoffs done by the CSMA-CA algorithm. Note that the range defined by the 802.15.4 standard is 0 to 5; however, the HW-MAC allows values up to 7.

Register: Battery Life Extension Window Length[3:0]

mnemonic: macBattLifeExtPeriods, address 0xBD, access R/W, C code-based constant address definition: CSMA_BATT_LIFE_EXT_PRDS

Addr.	Register	Description	Reset
0xBD	macBattLifeExtPeriods	Length of the backoff and transaction start window in back off periods in the Battery Life Extension mode	0x06

Reference: IEEE 802.15.4, Section 7.5

This register defines the number of backoff periods after the beacon IFS in which the backoff countdown or a transaction start can take place if the Battery Life Extension mode is enabled.

Register: CSMA Random Backoff Generator Seed [7:0]

mnemonic: CsmaSeed, address 0xBE, access R/W, C code-based constant address definition: CSMA_SEED

Addr.	Register	Description	Reset
0xBE	CsmaSeed	Seed value of the random backoff generator	0xAA

Writing to this register reinitializes the CSMA random backoff generator with a new seed.

5.15 SPI Registers

Register: SPI Configuration [5:0]

mnemonic: SPIconfig, address 0xFB, access: R/W, C code-based constant address definition: SPI_CONFIG

Addr.	Register	Bit	Description	Reset
0xFB	SPIconfig	0	SPR0	0
		1	SPR1	0
		2	CPHA	0
		3	CPOL	0
		4	MSTR	0
		5	SPE	1
		[7:6]	Not implemented	/

SPE - SPI System Enable
0 = SPI system is off.
1 = SPI system is on.

MSTR - Master/Slave Mode Select
0 = SPI is configured as a slave.
1 = SPI is configured as a master.

CPOL - Clock Polarity Select
0 = Active high clocks selected; SCK idles low.
1 = Active low clocks selected; SCK idles high.

CPHA - Clock Phase Select
Only CPHA = 0 mode is supported. Do not change this bit.

SPR1, SPRO - SPI Bit Rate Select (Master Mode)

The following table shows the relationship between the SPR1 and SPRO control bits and the bit rate for transfers when the SPI is operating as a master. When the SPI is operating as a slave, the serial clock is input from the master; therefore, the SPR1 and SPRO control bits have no meaning.

SPR1	SPRO	Master Mode SCK Rate	
		EU	US
0	0	1.50 MHz	3.00 MHz
0	1	0.75 MHZ	1.50 MHZ

Register: SPI Master Mode Start [0:0]

mnemonic: SPIstart, address 0xFC, access: R/W, C code-based constant address definition: SPI_START

Addr.	Register	Description	Reset
0xFC	SPIstart	SPI master transaction start command (start = 0x01)	0x00

This is for Master mode only (SPIconfig, MSTR=1). Writing 0x01 to it starts an SPI master transaction. The SPItx is shifted out on the MOSI line, and the received byte on the MISO line is stored in the SPIrx register. The start bit is automatically reset by hardware.

Register: SPI Transmit Byte [7:0]

mnemonic SPItx, address 0xFD, access: R/W, C code-based constant address definition: SPI_TX

Addr.	Register	Description	Reset
0xFD	SPItx	SPI Master mode transmit byte	0x00

This is for Master mode only (SPIconfig, MSTR=1). This register contains the SPI transmit byte. The SPItx byte is shifted out on the MOSI line after the SPIstart (0x01) command.

Register: SPI Receive Byte [7:0]

mnemonic SPIrx, address: 0xFE, access: R, C code-based constant address definition: SPI_RX

Addr.	Register	Description	Reset
0xFE	SPIrx	SPI Master mode receive byte	0x00

This is for Master mode only (SPIconfig, MSTR=1). At the end of an SPI master transaction started by the SPIstart (0x01) command, the SPIrx register contains the data read from the MISO line.

5.16 CLKO Configuration

The ZWIR4501 can provide a clock signal on its CLKO pin. It is enabled by default.

Register: CLKO Output Configuration [7:0]
mnemonic: ClkOutConfig, address 0xB8, access R/W, C code-based constant address definition: CLK_OUT_CONFIG

Addr.	Register	Bit	Description	Reset
0xB8	ClkOutConfig	[1:0]	SleepModeClock	1
		[3:2]	NormalModeClock	2
		[5:4]	Clk24Div, N	2
		[7:6]	RtcDiv, M	0

This register configures the clock output on the CLKO pad. The clock output depends upon the operation. It can be set up differently for the Sleep/Global Power Down mode and the Normal mode. By default, the 24 MHz divided by 4 is selected in Normal mode, and the 32.768 kHz RTC clock is selected in Sleep/Global Power Down mode.
Note that if the CLKO is configured to drive a clock derived from the 24 MHz in Sleep or Global Power Down mode, the 24 MHz crystal oscillator is not powered down and therefore the low Sleep/Global Power Down current consumption is not achieved in these modes.

SleepModeClock – clock configuration in Sleep and Global Power Down mode
0 = OFF
1 = 32.768 kHz / M (default configuration)
2 = 24 MHz / N

NormalModeClock – clock configuration in other than Sleep or Gobal Power Down mode
0 = OFF
1 = 32.768 kHz / M
2 = 24 MHz / N (default configuration)

Clk24Div, N – 24 MHz clock divider
0 = divide by 1
1 = divide by 2
2 = divide by 4 (default configuration)
3 = divide by 8

RtcDiv, M – 32.768 kHz clock divider
0 = divide by 1 (default configuration)
1 = divide by 2
2 = divide by 4
3 = divide by 8

To reduce the power consumption, it is recommended that the clock output signal be switched off if it is not in use.

5.17 Recommended Startup Register Setup

Register: Multiple Times Programmable (MTP) Memory Control [3:0]

mnemonic: MTPcontrol, address 0x37, access R/W, C code-based constant address definition: MTP_CTRL

Addr.	Register	Description	Reset
0x37	MTPcontrol	MTP control register	0x00

This register is used to overwrite the values of the trimming register and the transceiver's unique 64-bit source address register mhrSrcAddr64Tx in the register bank with the values stored in the MTP. The MTP is programmed by ZMD during production. The mhrSrcAddr64Tx address in the register bank is replaced by the programmed IEEE 64-bit extended unique identifier (EUI-64). ZMD's 24-bit OUI is 0x00117D.

Recommended MTP procedure:

- Write the value 0x09 to the MTPcontrol register to initiate the setting
- Wait for 500 µs or poll the MTPcontrol register by reading it and wait until the read register value's is equal to 0x0C
- Write 0x00 to the MTPcontrol register to complete the setting

Note that the MTP procedure can be run only once after reset.

After running the MTP procedure, it is recommended that the following registers be set for best performance results:

- Write to register address 0x39 (CLIP_CHK_AGC_LVL_TH) the value 0x50
- Write to register address 0x0F (AGC_LVL_INIT) the value 0x90
- Write to register address 0x2E (ACQ_PEAK_TH_SC_TRIM) the value 0xC3
- Write to register address 0x1D (RSTX) the value 0x0B
- Write to register address 0x22 (ED0_TRIM) the value 0x96

In addition to the above, for EU mode:

- Write to register address 0x3A (CLIP_CNT_STARTSMP) the value 0xF0
- Write to register address 0x2B (ACQ_PEAK_TH_1) the value 0x03
- Write to register address 0x2A (ACQ_PEAK_TH_0) the value 0xA0

or for US mode:

- Write to register address 0x08 (RX_MODE) the value 0x62
- Write to register address 0x3A (CLIP_CNT_STARTSMP) the value 0x78
- Write to register address 0x3B (CLIP_CNT_TH) the value 0x04
- Write to register address 0x2B (ACQ_PEAK_TH_1) the value 0x04
- Write to register address 0x2A (ACQ_PEAK_TH_0) the value 0x00

Check note of the RPCC register regarding channel 10.

To reduce the power consumption, it is recommended that the clock output signal be switched off. Refer to section 5.16 for further information on the ClkOutConfig register settings.

If the ZWIR4501 is used with an external power amplifier, see section 3.3 for further required startup register setup.

Note:
Please contact ZMD to receive the latest values before using the recommended values in a product release.

6 Transmitter RF Spectrum Test Modes

The transmitter RF spectrum can be measured on the RFIO or RFO pin, depending upon the TXIO mode bit in the RTXM register. It is highly recommended that the MTP procedure be run to ensure the best RF performance before the transmitter RF spectrum test is initiated.

For output of a repetitive pn-code sequence (as is used in the preamble):

- Set the RTXM = 0x04 for RFIO or RTXM = 0x0C for RFO
- Set Direct Tx mode (0x04) in the macTxConfig register
- Start the transmitter by writing TxOn (0x03) to the macControl register

A typical pn-code sequence signal is displayed by below. From this configuration, the test mode pattern from pn-code sequence can be changed to Continuous Carrier Output mode by setting RTXM = 0x02. A typical continuous carrier signal is displayed below.

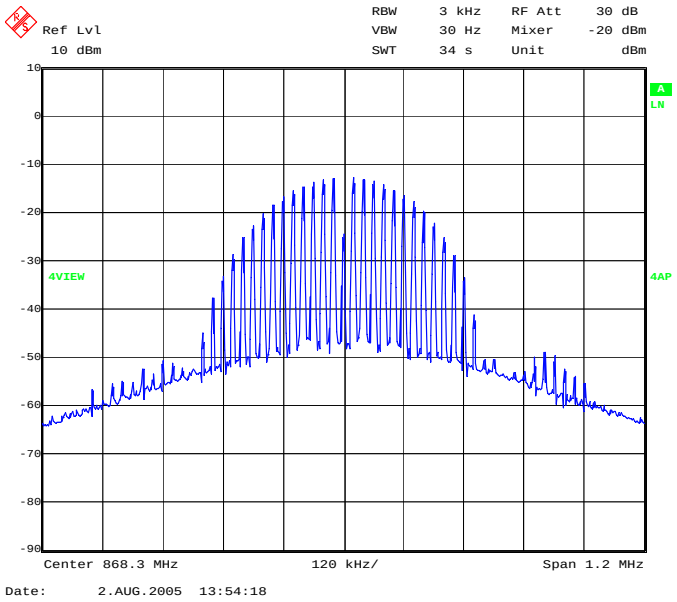


Figure 6.1: Typical pn-Code Sequence Signal

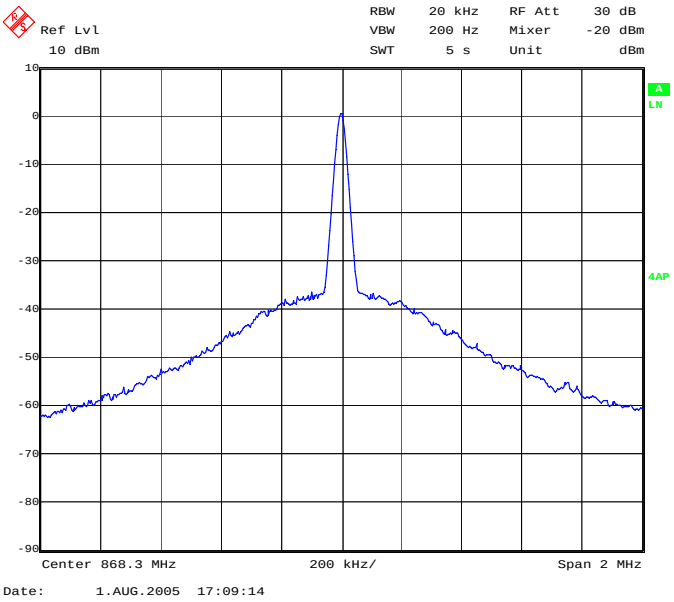


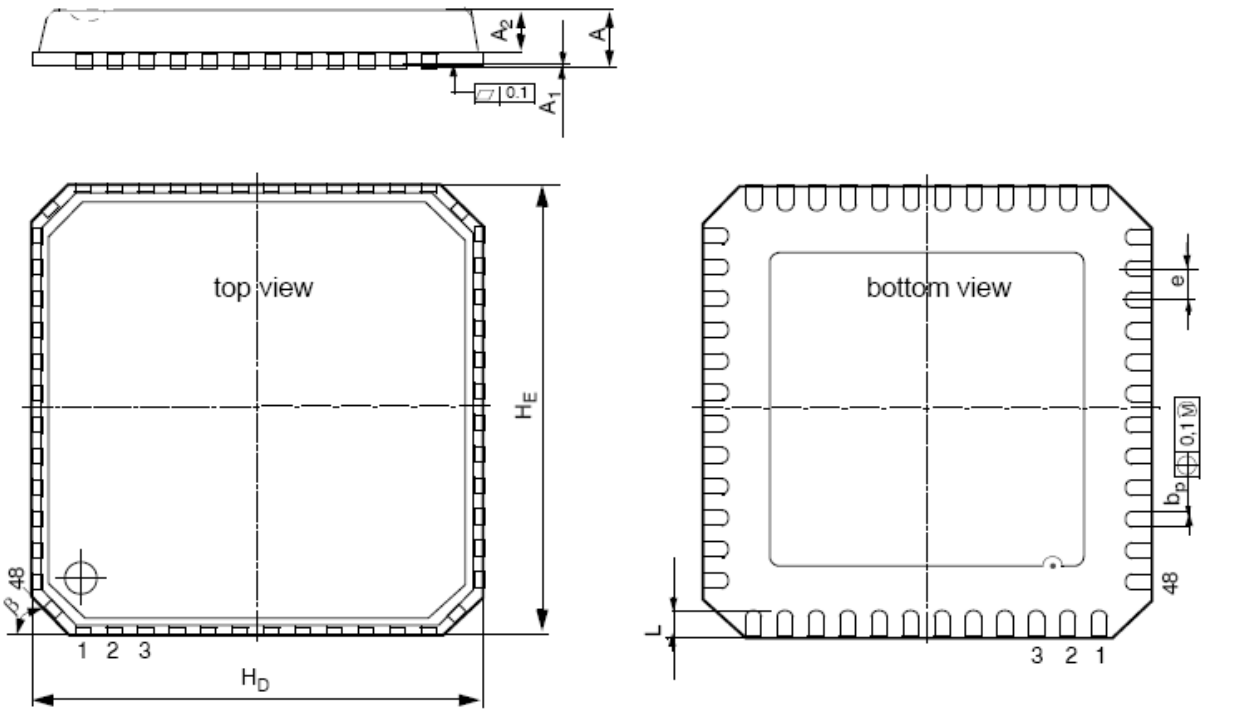
Figure 6.2: Typical Continuous Carrier Signal

Changing the RFIO to RFO or vice versa can be done only by running the sequence described above again from the beginning and selecting the required mode in the first step.

7 Mechanical Specifications

The ZWIR4501 uses a green package (RoHS). Package type: MLF (PQFN48) (7 x 7)

1 Dimensions



Dimensions of Sub-Group B1	
e_{nom}	0.5
A_{max}	0.9
b_{Pmin}	0.18
b_{Pmax}	0.30
H_{Emin}	6.9
H_{Emax}	7.1
H_{Dmin}	6.9
H_{Dmax}	7.1
L_{min}	0.3

Dimensions of Sub-Group C1	
A_{min}	-
A_{1min}	0.0
A_{1max}	0.05
A_{2min}	-
A_{2max}	0.7
β	45°

2	**Weight**	≤ 0.14 g
3	**Package Body Material**	Low Stress Epoxy
4	**Lead Material**	Cu-Alloy
5	**Lead Finish**	solder plating
6	**Lead Form**	no lead

8 Limitations

This section gives an overview of the currently known issues with the ZWIR4501 samples.

Issue	Short Description / Reason	Temporary Workaround Until New Release
Higher PER	Selectivity issues at high sensitivity	Use latest version of the "Recommended Startup Register Setup." See section 5.17. *)
Detection of frames in adjacent channel during scan	A crosstalk from an adjacent channel could cause a misinterpretation of scan result.	Use RSSI value as additional information for right frame detection.

*) Contact ZMD's support team (zigbee@zmd.de) for further information.

9 Document Revision History

Revision	Description	Release date
1.0	First official release (ZMD44102)	25 Mar 2006
1.1	Changes to version 1.0 (besides minor corrections): Update sections "1.4 Pin Assignment" and "1.5 Pin Description" with description of ATESTx, RFTEST and RFTESTPWR pins; Correct DirectFifoAccess Mode description regarding mhrAckFc1Tx register (section "4.5 Frame Handling"); Update "Off mode" description in section "4.11.1 Low Power Modes"; Rearrange section "4.11.1 Low Power Modes"; Update section "4.6 Link Quality Indicator (LQI)" regarding use of AgcLvl value instead of LQI value that is stored in the RxFIFO; Recommend to use the AgcLvl value to calculate the LQI value instead of the two bytes that are stored after a received frame to the RxFIFO; Update section "4.7 Receive Signal Strength Indicator (RSSI)" regarding curve and approximation equation; Update section "4.8 Energy Detection Level" regarding curve and approximation equation; Correct section "5.17 Recommended Startup Register Setup" regarding recommended setting of the ED0_TRIM register value;	28 Aug 2006
1.2	Add information about the order of the scan mode initialization sequence in section "4.11.4 Scan Modes"; Update section "4.5 Frame Handling" and description of the T_SIFS and the T_LIFS register regarding limitations in DirectFifoAccess mode;	4 Jun 2007

10 List of Abbreviations

ADC	Analog-to-Digital Converter
AES	Advanced Encryption Standard
AGC	Automatic Gain Control
BER	Bit Error Rate
BPSK	Binary Phase Shift Keying
CMOS	Complementary Metal Oxide Silicon
CRC	Cyclic Redundancy Check
CSMA	Carrier Sense Multiple Access
DAC	Digital-to-Analog Converter
dB	Decibel
DSSS	Direct Sequence Spread Spectrum
ED	Energy Detection
ESD	Electrostatic Discharge
ETSI	European Telecommunications Standards Institute
EU	Europe
FCC	Federal Communications Commission
FIFO	First In First Out
GPD	Global Power Down
GTS	Guaranteed Time Slot
IEEE	Institute of Electrical and Electronics Engineers
IF	Intermediate Frequency
IFS	Interframe Spacing
IRQ	Interrupt Request
ISM	Industrial Scientific Medical
kbit/s	Kilobit per second
kHz	Kilohertz
LNA	Low Noise Amplifier
LoS	Line of sight
LP Filter	Low Pass Filter
MAC	Medium Access Controller
MISO	Master-In-Slave-Out,
MOSI	Master-Out-Slave-In
MHz	Megahertz
MLF	Micro Lead Frame
PER	Packet Error Rate
PHY	Physical (Layer)
PLL	Phase Locked Loop
QFN	Quad Flat No-Lead
QFP	Quad Flat Pack
RF	Radio Frequency
RSN	Reset Not
RTC	Real Time Clock
Rx	Receiver, Receive
SPI	Serial Peripheral Interface
SSN	Slave-Select Not (refers to CS=Chip Select)
Tx	Transmitter, Transmit
US	United States
XTAL	Crystal

11 References

[1] IEEE Std. 802.15.4-2003: “IEEE Standard for Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (LR-WPANs)” Download: http://standards.ieee.org/getieee802/download/802.15.4-2003.pdf

[2] www.zmd.biz

[3] www.etsi.org

[4] www.fcc.gov

[5] ZWIR4501 Starter Kit User Guide including Basic Communication Software; software code examples controlling the ZWIR4501

[6] C code header file that contains the ZWIR4501’s register address and other defines like commands, status and configuration settings; can be requested from zigbee@zmd.de

[7] ZWIR4501 Application Note 01: RF Reference Design incl. PCB Design Guidelines; ZWIR4501_AN01_RF_Ref_Design_v1_0.zip, can be requested from zigbee@zmd.de

[8] ZWIR4501 Application Note 07: Hardware Abstraction Layer (HAL) for C8051F12x ZWIR4501_AN07_HAL_C8051F12x_v1_0.zip

[9] ZWIR4501 Software MAC Layer, implementation of a SW-MAC layer for the ZWIR4501 Starter Kit board; can be requested from zigbee@zmd.de

Disclaimer

The information furnished herein by ZMD is believed to be correct and accurate as of the publication date. However, ZMD shall not be liable to any third party for any damages, including but not limited to personal injury, property damage, loss of profits, loss of use, interruption of business, or indirect, special, incidental, or consequential damages of any kind in connection with or arising out of the furnishing, performance, or use of the technical data. No obligation or liability to any third party shall arise from ZMD's rendering technical or other services.

Products sold by ZMD are covered exclusively by the ZMD standard warranty, patent indemnification, and other provisions appearing in ZMD standard "Terms of Sale". Testing and other quality control techniques are used to the extent ZMD deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed. ZMD makes no warranty (express, statutory, implied and/or by description), including without limitation any warranties of merchantability and/or fitness for a particular purpose, regarding the information set forth in the Materials pertaining to ZMD products, or regarding the freedom of any products described in the Materials from patent and/or other infringement.

ZMD reserves the right to discontinue production and change specifications and prices, make corrections, modifications, enhancements, improvements and other changes of its products and services at any time without notice. ZMD products are intended for use in commercial applications. Applications requiring extended temperature range, unusual environmental requirements, or high reliability applications, such as military, medical life-support or life-sustaining equipment, are specifically not recommended without additional mutually agreed-upon processing by ZMD for such applications. ZMD assumes no liability for application assistance or customer product design. Customers are responsible for their products and applications using ZMD components.

For further information contact

ZMD AG
Grenzstrasse 28
D-01109 Dresden
Germany
Phone: +49 351 88 22 0
Fax: +49 351 88 22 606
zigbee@zmd.de

ZMD America
15373 Innovation Drive
Suite 115
San Diego, CA 92128
Phone: +1 858 674 8070
Fax: +1 858 674 8071
wireless@zmda.com

ZMD Far East
1F, No. 14, Lane 268
Sec. 1 Guangfu Road
Hsinchu City 300
Taiwan
Phone: +886 (0) 3 563 1388
Fax: +886 (0) 3 563 6385

General Information
http://www.zmd.biz

Support Contact
wireless_support@zmda.com

Print date: 8/27/2009 11:40:00 PM